



UNIVERSIDADE D  
COIMBRA

Joel Alves Costa Filho

**FEASIBILITY STUDY OF USING A SATELLITE  
STAR TRACKER FOR SPACE DEBRIS DETECTION  
ON LEO ORBITS**

**Dissertation in the context of the Master in Astrophysics and Instrumentation for  
Space oriented by Doctor Paulo Gordo, Doctor Nuno Peixinho and Doctor Ricardo  
Gafeira and presented to the Department of Physics of the University of Coimbra**

July 2021

**“You cannot hope to build a better world without improving the individuals. To that end, each of us must work for his own improvement and, at the same time, share a general responsibility for all humanity, our particular duty being to aid those to whom we think we can be most useful.”**

*Marie Curie*

## *Acknowledgments*

First of all, I would like to thank the coordinates of my course for the opportunity to do this thesis. I also like to thank the OGAUC - Geophysical and Astronomical Observatory of the University of Coimbra for providing all the structure to this work. For family, friends and my advisors, the acknowledgements will be in Portuguese.

Antes de tudo agradeço a Deus por todas as oportunidades. Joel e Heloisa, mais conhecidos como pai e mãe, não estaria aqui sem vocês. Portanto, agradeço do fundo do meu coração todo o amor, suporte e apoio dado para que eu consiga alcançar meus sonhos. Além disso, também agradeço aos meus irmãos, Ana Mônica e José Alberto, e meus avós.

Irvylle, obrigado por estar sempre ao meu lado e por ser meu porto seguro. A vida ficou mais fácil com você ao meu lado. Aos meus amigos Thamyres, Jônatas, Luiz, Jade, Marquinhos, Martinha e João Marcos o meu muito obrigado. Minha experiência em Portugal não seria a mesma sem vocês. Também agradeço a todos meus amigos e familiares do Brasil que mesmo de longe continuam sendo importantíssimos para mim.

Também agradeço imensamente aos meus orientadores. Ao doutor Paulo Gordo, muito obrigado por sugerir essa tese e sempre acompanhar o desenvolvimento do trabalho mesmo a distância. Ao Doutor Ricardo Gafeira, muito obrigado por me ajudar a entender melhor os sensores e algoritmos utilizados. E por último, mas não menos importante, ao doutor Nuno Peixinho. Muito obrigado por ter sido incansável, paciente e estar sempre ao meu lado me dando todo o apoio que necessitava.

Por fim, agradeço a todos que direta ou indiretamente contribuíram para que eu chegasse até aqui. Muito obrigado!

# *Abstract*

Space activities contribute significantly to science and technology. However, the increase in missions has created the problem of space debris, which are nonfunctional objects in orbit. The large concentration of debris represents a risk, so it is of utmost importance to detect it. This thesis studies the feasibility of using a star tracker to detect debris in *LEO*. Star tracker is a device that uses star imaging to extract the satellite attitude. Thus, a literature review was done to know the size, distribution and magnitude of the debris and study the star trackers' architecture. In addition, several algorithms for the detection of stars and debris were searched. The MT9J001 sensor was chosen for the optical system, and noise characterization was reviewed. The FOVs were calculated, and a value of 7 was determined for the limiting magnitude. The star detection algorithm chosen was Tetra, which can detect patterns of 3 or 4 stars and then calculate the attitude accurately. A change was made to replace the original catalog with the one with more stars. Then, Tests were made with both catalogs. For detecting debris, ASTRiDE was chosen, an algorithm that uses information from the edges of objects to differentiate stars from debris for identification. Tests were made with several images from the Internet. In order to calculate the probability of debris detection, first, the debris size was obtained as a function of the threshold magnitude. Then the distribution of objects on the celestial sphere was calculated. Finally, the detection probability was determined for a field of view of  $1\text{deg}^2$  and for the FOV of the system on the celestial sphere and specific orbit regions. At last, photographs were taken using the telescope and lens to test the system.

*Keywords:* space debris, star tracker, attitude, probability, detection

## ***Resumo***

As atividades espaciais contribuem significativamente para a ciência e tecnologia. Entretanto o aumento das missões gerou o problema dos detritos espaciais, os quais são objetos sem funcionalidade em órbita. A grande concentração de detritos representa um risco, por isso é de suma importância detetá-los. A presente tese estuda a viabilidade de utilizar um star tracker para detetar detritos na *LEO*. Star tracker é um dispositivo que utiliza imagem de estrelas para extrair a atitude do satélite. Para tanto, foi feita uma revisão da literatura para saber o tamanho, distribuição e magnitude dos detritos e, também, estudar a arquitetura dos stars trackers. Ademais, foram pesquisados diversos algoritmos para a detecção de estrelas e detritos. Escolheu-se o sensor MT9J001 para o sistema óptico e caracterização do ruído foi revisada. Os FOVs foram calculados e determinou-se um valor de 7 para a magnitude limite. O algoritmo de detecção de estrelas escolhido foi o Tetra, o qual consegue detetar padrões de 3 ou 4 estrelas e calcular a atitude com precisão. Uma alteração foi feita para trocar o catálogo original por outro com mais estrelas. Testes foram feitos com os dois catálogos. Para detetar detritos, escolheu-se o ASTRiDE, algoritmo que utiliza informações das bordas dos objetos para diferenciar estrelas de detritos e identificá-los. Realizou-se testes com várias imagens da internet. Com o intuito de calcular a probabilidade de detecção de detritos, primeiro obteve-se o tamanho dos mesmo em função da magnitude limite. Depois calculou-se a distribuição de objetos na esfera celeste. Por fim, determinou-se a probabilidade de detecção para um campo de visão de  $1deg^2$  e para o FOV do sistema em toda a esfera celeste bem como para regiões específicas. Finalmente, fotografias foram feitas utilizando o telescópio e a lente para testar o funcionamento do sistema.

*Palavras-chave:* detritos espaciais, star tracker, atitude, probabilidade, detecção

# Summary

1	Introduction .....	1
1.1	Motivation and Goals .....	1
1.2	Thesis Content .....	2
2	Space Debris .....	3
2.1	Types of orbit .....	3
2.2	Types and distribution of space debris .....	3
2.3	The main coordinate systems used for debris .....	7
2.4	The visual magnitude of debris .....	9
2.5	Algorithms to detect space debris .....	11
3	Star Tracker .....	14
3.1	Architecture and operation of a star tracker .....	14
3.2	Image sensor .....	15
3.2.1	Astronomical photometry .....	17
3.2.2	Signal .....	18
3.2.3	Noise analyses .....	19
3.3	Star detection algorithms .....	20
3.4	Market survey .....	22
4	Optical system .....	24
4.1	Image sensor .....	24
4.2	Sensor characterization .....	25
4.2.1	Charge conversion efficiency .....	25
4.2.2	Readout noise .....	26
4.2.3	Dark noise .....	27
4.3	Optical system FOV .....	27
4.4	Prediction of the limiting magnitude .....	28
5	Algorithms .....	32
5.1	Tetra algorithm .....	32
5.1.1	Internal catalog .....	33
5.1.2	Output .....	33
5.1.3	Change in the internal catalog .....	34
5.1.4	Tests with BSC5 and Hipparcos .....	35
5.2	Debris detection algorithm .....	37
5.2.1	ASTRiDE .....	37

5.2.2	Output .....	38
5.2.3	Tests.....	38
6	Probability of space debris detection.....	43
6.1	Using visual magnitude to determine the size of debris.....	43
6.2	Power law .....	45
6.3	Probability of space debris detection .....	46
6.3.1	Probability of detection in a specific orbit.....	48
6.3.2	Probability of detection between latitudes.....	50
7	Tests and results .....	54
8	Conclusions .....	55
9	References .....	56

# List of Figures

Figure 1 Tracked objects in space (ESA, 2020a) .....	4
Figure 2 Altitude distribution in LEO. Each bin has a 25 km altitude (Bonnal & McKnight, 2017) .....	6
Figure 3 Inclination and altitude distribution in LEO. Each bin has 2° and 50 km (Bonnal & McKnight, 2017) .....	6
Figure 4 Altitude and inclination in terms of spatial density. Results obtained by using the MASTER model (ESA, 2009) .....	7
Figure 5 Schematic of the three coordinate systems (Lopes, 2018).....	8
Figure 6 Phase angle: Sun-object-observer angle (Kervin et al., 2010).....	10
Figure 7 Debris, stars, and noise in the time- index image (Xi et al., 2016).....	11
Figure 8 Example of a perfect match with the filter proposed (Vananti et al., 2015)....	12
Figure 9 Segmentation process (Virtanen et al., 2016) .....	13
Figure 10 Example of a typical star tracker. Adapted from (Sarvi et al., 2020) .....	14
Figure 11 Structure of a CCD showing that the image is read line by line. Adapted from (Waltham, 2013).....	16
Figure 12 CMOS array of pixels. Adapted from (Waltham, 2013).....	16
Figure 13 Point spread function of a star with the format of an Airy diffraction disk (Lopes, 2018).....	18
Figure 14 MT9J001 connected with the Arducam USB2 Camera Shield through a 30pin ribbon cable .....	24
Figure 15 Integration sphere and experimental setup did by Lopes (Lopes, 2018) .....	26
Figure 16 Optical system: lens + sensor.....	27
Figure 17 Optical system: sensor + telescope .....	28
Figure 18 Johnson-Cousins system filter curves. Adapted from (Lopes, 2018) .....	29
Figure 19 MT9J001 spectral sensitivity curve. Available in the sensor datasheet.....	30
Figure 20 Approximation in the Johnson-Cousins system to measure the flux. Adapted from (Lopes, 2018).....	30
Figure 21 SNR for different magnitudes in distinguished integration time. ....	31
Figure 22 Coordinate system used by Tetra (Brown et al., 2017).....	33
Figure 23 On the left, there is an image of the sky taken from the internet. On the right, it is the same image after being analyzed by Tetra.....	34
Figure 24 Short: original image.....	39
Figure 25 Short: output from ASTRiDE. The streak was identified as 1 .....	39
Figure 26 Long: original image.....	40
Figure 27 Long: output from ASTRiDE. Two streaks were identified and denominated as 1 and 2.....	41
Figure 28 Visual magnitude for three different mixing coefficients for a 10 cm debris in an 800 km orbit.....	43
Figure 29 Polar orbit (ESA, 2020b) .....	49
Figure 30 Spherical coordinates .....	49
Figure 31 A hundred different debris trajectories in an orbit of latitude between -80 and 80 deg.....	51
Figure 32 Histogram for a thousand different debris trajectories in an orbit of latitudes between -80 and 80 deg .....	52
Figure 33 Probabilities of finding objects inside a band of 2 deg for different latitudes .....	52

## *List of Tables*

Table 1 Space debris classification by size. Adapted from (Andrenucci et al., 2011).....	5
Table 2 Survey of star trackers .....	23
Table 3 Characteristics of the MT9J001 10 Mp CMOS digital image sensor .....	24
Table 4 Flux calibration values of A0 stars for the Johnson-Cousins system. Adapted from (Lopes, 2018).....	29
Table 5 Results for right ascension .....	35
Table 6 Results for declination.....	35
Table 7 Results for Roll values.....	36
Table 8 Results for FOV.....	36
Table 9 Results for the mismatch probability.....	36
Table 10 Streak parameters .....	40
Table 11 Streaks 1 and 2 parameters.....	41
Table 12 Comparison of the satellites' simulated and observational magnitudes.....	44
Table 13 Number of debris for different sizes at the reference epoch of May 2009. Adapted from (Bonnal & McKnight, 2017) .....	46
Table 14 Detection probabilities of space debris in 800 km orbit for different exposure times and velocities .....	47
Table 15 Detection probability using the FOV of the image sensor in the telescope ....	48
Table 16 Detection probability in the most crowded area using the FOV .....	50
Table 17 Detecting probability for at least one object in latitudes between 78 and 80 deg .....	53

## *Abbreviations and Acronyms*

<b>ADU</b>	Analog-to-Digital Unit
<b>Al<sub>2</sub>O<sub>3</sub></b>	Aluminum oxide
<b>AU</b>	Astronomical unit
<b>BSC5</b>	Bright star catalog
<b>CCD</b>	Charge Couple Device
<b>CCE</b>	Charge Conversion Efficiency
<b>cm</b>	Centimeter
<b>CMOS</b>	Complementary metal-oxide-semiconductor
<b>db</b>	Decibels
<b>deg</b>	Degrees
<b>ECEF</b>	Earth-Centered Earth Fixed coordinate
<b>ECI</b>	Earth-Centered Inertial coordinate
<b>ESA</b>	European Space Agency
<b>FOV</b>	Field of view
<b>GEO</b>	Geostationary orbit
<b>ID</b>	Identity
<b>ISS</b>	International Space Station
<b>kg</b>	Kilogram
<b>km</b>	Kilometer
<b>LEO</b>	Low Earth orbit
<b>m</b>	meter
<b>MASTER</b>	Meteoroid and Space Debris Terrestrial Environment Reference
<b>MEO</b>	Medium Earth orbit
<b>mg</b>	Milligram
<b>MHz</b>	Megahertz
<b>mm</b>	Millimeter
<b>ms</b>	Milliseconds
<b>Mp</b>	Megapixel
<b>OCS</b>	Orbital coordinate system
<b>PSF</b>	Point spread function

<b>QE</b>	Quantum efficiency
<b>s</b>	Seconds
<b>SI</b>	Standard International unit
<b>SNR</b>	Signal-to-noise ratio
<b>US</b>	United States
<b>USSR</b>	Union of Soviet Socialist Republics
<b>W</b>	Watt
<b>μm</b>	Micrometer

# 1 Introduction

In October 1957, the USSR placed in orbit the *Sputnik 1*, the first artificial object in human history, starting space exploration (Rossi, 2005). Over the years, space activities contribute to technological and scientific developments. However, the increase in the number of objects in orbit has led to a significant problem: space debris.

Space debris is any nonfunctional object in orbit, such as inactive satellites, rocket bodies, fragmentation objects. The first known fragmentation in orbit was in 1961 when the *Transient 4A* rocket body exploded. From then on, the repetition of launches of new satellites and the breakup of in-orbit spacecraft contributed to building up a considerable population of debris, which now represents a threat for all the space missions (Rossi, 2005).

To continue to operate missions in orbit safely, the observation and tracking of debris are essential. Usually, these are done by optical and radar telescopes in the ground. The purpose of this work is to study a different approach to debris observation using star trackers.

Star tracker is a device capable of extracting the spacecraft attitude with high accuracy using the stars in its field of view. This work proposes a dual-purpose star tracker feasibility study, which determines the attitude and identifies space debris.

## 1.1 Motivation and Goals

The growing population of space debris is a threat to the spacecraft in orbit and future missions. If the actual scenario does not change, the concentration of debris could reach unsustainable values for space exploration. The first step to modify this context is to detect and track as many as possible objects to take more active action, like removing debris from orbit in the near future. So, it is vital to develop new techniques to detect these objects besides the telescope's observations.

The main goal of this work is to study the feasibility of using star trackers for debris detection on *LEO* orbit. The low-cost star tracker developed by Lopes (2018) was used to achieve this goal. Then, it is proposed a system capable of extracting a satellite's attitude, employing a high sophisticated algorithm called Tetra, and detecting possible debris by an open-source algorithm denominated ASTRiDE.

## **1.2 Thesis Content**

This work is structured as follows:

### **Chapter 2**

This chapter is an overview of space debris, describing the types and distribution of these objects in orbit. Also, it is presented the coordinates system of space objects and the calculations of the debris visual magnitude. The chapter is finalized with a brief description of different algorithms to detect space debris.

### **Chapter 3**

The star tracker is studied in this chapter. It is started with the architecture of this device. Then the optical system is presented, describing the image sensors and the theory of astronomical photometry. The chapter also contains a summarized description of different star detection algorithms, and it ended with a market survey of star trackers.

### **Chapter 4**

The selected optical system is described in this chapter. It is presented the characterization of the sensor's noise and the calculations of the FOV and limiting magnitude of the sensor.

### **Chapter 5**

In chapter 5, the algorithms to extract the attitude, Tetra, and detect space debris, ASTRiDE, are described and tested. Tetra's internal catalog was changed, so the technique for this modification and the comparison between the original and updated versions were also presented.

### **Chapter 6**

Chapter 6 describes the probability of detecting space debris. It starts to calculate the visual magnitude of debris for a specific size. Then the distribution of these objects is determined by a power law. Finally, the detection probability is presented for different FOVs and regions.

### **Chapter 7**

### **Chapter 8**

## 2 Space Debris

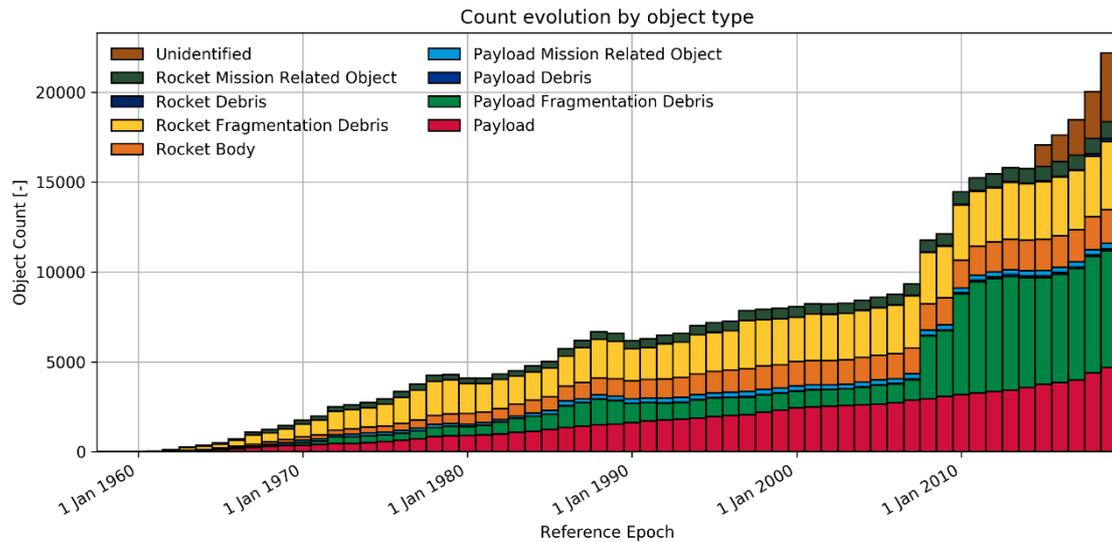
### 2.1 Types of orbit

Every object that is launched into space is placed in a specific orbit related to the purpose of the mission. The Earth space region can be divided into three main orbits, the Geostationary orbit, the Low Earth orbit, the Medium Earth orbit (ESA, 2020b), characterized as:

- Geostationary orbit (*GEO*): objects in this region circle the Earth above the equator following Earth's rotation, making the object appear 'stationary' over a fixed position. The *GEO* region is at an altitude of 35 786 km, and it is commonly used by telecommunication satellites.
- Low Earth orbit (*LEO*): is an orbit relatively close to the Earth's surface, ranging from 160 km to 2000 km. *LEO* is the most commonly used orbit to its proximity and to allow more available routes. It is the orbit used for satellite imaging and the International Space Station (*ISS*).
- Medium Earth orbit (*MEO*): comprises a wide range of orbits between *LEO* and *GEO*. Like *LEO*, *MEO* allows a great variety of paths around the Earth. It is commonly used by navigation satellites.

### 2.2 Types and distribution of space debris

There have been more than 5250 launches throughout space activities, resulting in some 42 000 tracked objects in space. About 23 000 continue in orbit and are, for instance, regularly tracked and held in a catalog by the US Space Surveillance Network (*US Space Surveillance Network*, n.d.). Only a tiny fraction of these objects, around 1200, are operational satellites (ESA, 2020a). The vast majority are space debris, also known as orbital debris or 'space junk'. Space debris is, by definition, any human-made object, or fragments or portions of it, in orbit around the Earth that no longer serve a useful function.



*Figure 1 Tracked objects in space (ESA, 2020a)*

As can be seen in Figure 1, there are a lot of different objects in space. Therefore, space debris can be classified as belonging to one of the four types of debris: nonfunctional spacecraft, rocket bodies, mission-related debris, and fragmentation debris (Culp, 1991). Nonfunctional spacecraft represent all the spacecraft that reach the end of life and remain in orbit. The vehicle that places these objects in orbit leaves behind one or more stages in space, known as rocket bodies. Usually, only one rocket body is released in LEO missions, while launching a vehicle for higher orbits, such as GEO, can drop up to three stages (ESA, 2020a).

Mission-related debris is released due to the spacecraft's deployment, activation, and operation, such as protective covering from the payload. Another kind of this type of debris is related to the operation of solid rocket motors. During the burning process, many aluminum oxide ( $\text{Al}_2\text{O}_3$ ) particles are formed and ejected from the rocket at velocities up to 4 km/s. While these particles' orbital life is relatively short, a considerable average population is observed by examining impacts on exposed spacecraft surfaces (ESA, 2020a).

The greatest amount of space debris can be classified as fragmentation debris. This type consists of objects created during breakups and deterioration. Breakups are destructive events, like internal explosions or collisions, that generate many smaller objects with a wide range of initial velocities. Deterioration is due to the harsh space environment, such as atomic oxygen, radiation. The products of this usually have a low relative velocity (ESA, 2020a).

From the objects in the catalog, about 50% are fragmentation debris (Andrenucci et al., 2011). Space debris can also be classified by its size: large, with a diameter greater than 10 cm and mass higher than 1 kg; medium, with a diameter from 1 mm to 10 cm and mass between 1 mg to 1 kg; and small, which have diameter lower than 1 mm and mass lesser than 1 mg. Table 1 shows more information about each type of this classification.

Table 1 Space debris classification by size. Adapted from (Andrenucci et al., 2011)

<b><i>Size</i></b>	<b><i>Number in orbit</i></b>	<b><i>Traceable</i></b>	<b><i>Lethal to Operational Spacecraft</i></b>	<b><i>Produce Lethal Fragment after Impact</i></b>
<b><i>Small</i></b>	~ 128 million	No	(Usually) Not	No
<b><i>Medium</i></b>	~ 900 000	No	Usually	Maybe
<b><i>Large</i></b>	~ 34 000	Yes	(Almost) Always	Yes

As can be seen in Table 1, traceable objects have limitations related to size. The US Space Surveillance Networking’s catalog has a size threshold ranging from 10 cm to *LEO* orbit to about 1 m in the *GEO* orbit. The debris observations are performed by a radar network for *LEO* and low *MEO* altitudes and optical telescopes for the remaining part of *MEO* up to *GEO*. The non-traceable objects are obtained statistically from debris environmental models (Bonnal & McKnight, 2017).

From the catalog objects, the vast majority is in *LEO*, more than 70%. About 4% is located in the geostationary ring (*GEO*). The remaining objects are spread across *MEO* and *GEO* transfer orbit (Bonnal & McKnight, 2017).

Considering only the *LEO* orbit population, Figure 2 shows the distribution of the objects in altitude. It is possible to see three maximum locations, the first one from 700 to 875 km, the second close to 975 km, and the third from 1400 to 1500 km (Bonnal & McKnight, 2017).

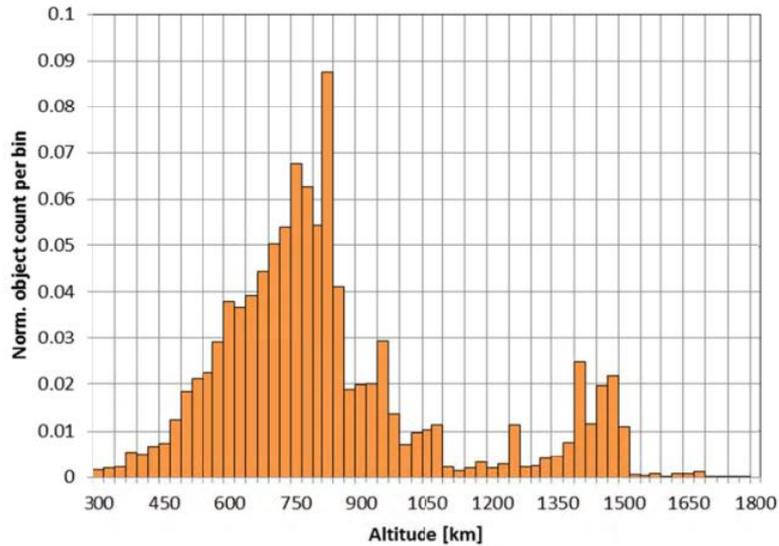


Figure 2 Altitude distribution in LEO. Each bin has a 25 km altitude (Bonnal & McKnight, 2017)

The distribution of orbital inclination among *LEO* orbits has a mission dependence, in which higher inclination is more common. The vast majority of objects are in inclinations between 95 to 100 deg (possessing inclinations above 90 deg means, by definition, that the orbit is retrograde) and from 70 to 80 deg. Fig 3 illustrates the histogram with the distribution of objects per orbital altitude and orbital inclination on LEO (Bonnal & McKnight, 2017).

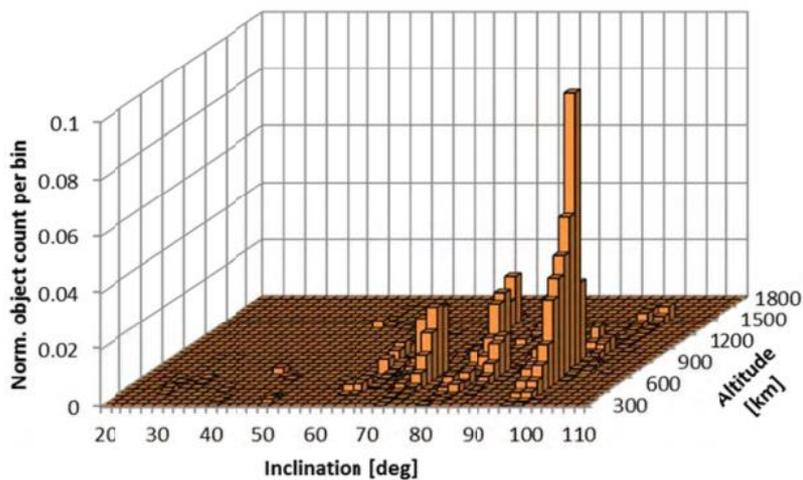


Figure 3 Inclination and altitude distribution in LEO. Each bin has 2° and 50 km (Bonnal & McKnight, 2017)

This figure refers to catalog objects, which represents a tiny part of the total. The only way to evaluate the distribution that also includes the non-catalog objects is by using computational models. One of these is the ESA’s MASTER (Meteoroid and Space Debris Terrestrial Environment Reference) model, which characterizes the space debris and meteoroid environment. The model’s purpose is the realistic description of natural and

artificial particles environments. The model gives the flux predictions on the user-defined target orbit. It covers a size regime for debris and micrometeoroids between 1 micrometer and 1 m. For debris, the model simulates the generation of objects due to all known sources of space junk (ESA, 2009).

Figure 4 shows the MASTER results for the distribution among the *LEO* population in terms of spatial density. The output illustrates the same behavior as the previous figures.

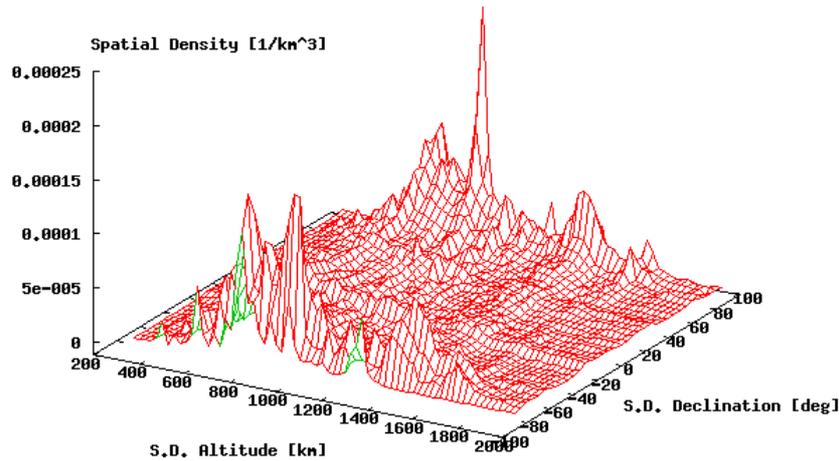


Figure 4 Altitude and inclination in terms of spatial density. Results obtained by using the MASTER model (ESA, 2009)

### 2.3 The main coordinate systems used for debris

The debris is tracked and identified using its position and orientation. These two values are defined by coordinate systems. The most common coordinate systems for objects in the Earth orbit, such as debris and satellites, are the Earth-Centered Inertial coordinate (*ECI*), Earth-Centered Earth Fixed coordinate (*ECEF*), and Orbital coordinate system (*OCS*) schematically shown in Figure 5 and briefly described below (e.g., Groves, 2013). The Earth-Centered Inertial coordinate has the origin centered at the Earth’s center of mass, and it is oriented concerning the Earth’s rotation axis and the stars. Inertial is a term used for any coordinate system that does not accelerate or rotate regarding the rest of the Universe. Since the Earth experiences acceleration and rotation, *ECI* is not strictly an inertial coordinate. However, these effects are smaller than the measurement noise of the navigation sensor so that *ECI* can be treated for practical purposes as an inertial frame (Groves, 2013).

In the *ECI* coordinate, the  $Z_{eci}$  points along the Earth’s rotation axis from the origin to the true north pole. The  $X_{eci}$  and  $Y_{eci}$  lie in the equatorial plane and do not rotate with the

Earth. The  $X_{eci}$  is in the direction from the Earth to the Sun in the vernal equinox and spring equinox in the northern hemisphere. This is the same direction from Earth's center to the intersection of the Earth's equatorial plane and the ecliptic (Earth-Sun orbital plane). Thus, *ECI* is sometimes known as celestial coordinates.

The Earth-Centered Earth Fixed coordinate is similar to *ECI*. Both share a common origin and have the same z-axis ( $Z_{eci} = Z_{ecef}$ ) but all the axes remain fixed concerning the Earth. The  $X_{ecef}$  is directed from the center to the intersection of the equator plane with the International Reference Meridian, which defines  $0^\circ$  longitude. The  $Y_{ecef}$  points from the center to the intersection of the equator with the  $90^\circ$  east meridian. *ECEF* is not used in this work, but it is important in navigation, allowing convert celestial coordinates to *GPS* (Groves, 2013).

The Orbital coordinate system has the origin placed in the center of mass of the Earth-orbiting object. The  $Z_{orb}$  points directly to the Earth's center of mass, the origin of the *ECI* coordinates. The  $X_{orb}$  has the same direction as the object's angular velocity. The  $Y_{orb}$  is perpendicular to the orbital plane and, of course, orthogonal to the other axes (Lopes, 2018).

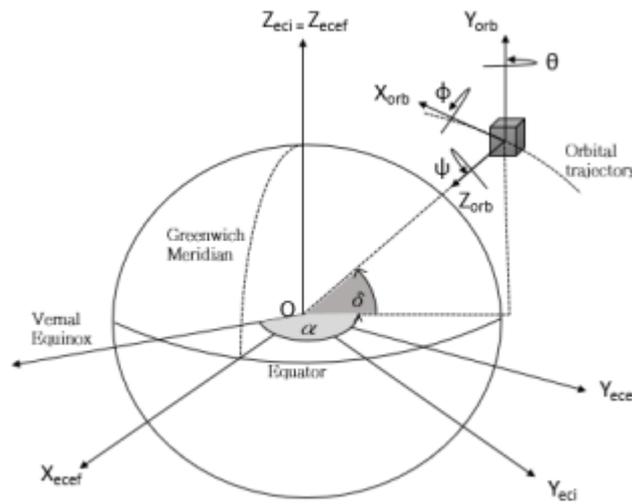


Figure 5 Schematic of the three coordinate systems (Lopes, 2018).

In Figure 5, it is represented the three coordinate systems and some relevant angles.  $\alpha$  is the right ascension, measured from the vernal equinox, and  $\delta$  is the declination, measured from the equator. The *OCS* coordinate is related to the Euler angles, which indicate the object's rotation regarding the coordinate axis. Each angle has a specific rotation movement,  $\phi$  is the roll,  $\theta$  is the pitch, and  $\psi$  is the yaw. In a satellite, these angles are

essential to determine the attitude, which describes the spacecraft's orientation relative to its center of mass (e.g., Groves, 2013)

## 2.4 The visual magnitude of debris

In astronomy, the apparent magnitude is a brightness scale for celestial bodies (planets, asteroids, stars, galaxies, etc.) as seen by an observer. Since the observer is accustomed to detect light in the visible range in optical telescopes, unless stated otherwise, the apparent magnitude is also known as visual magnitude. The visual magnitude is measured on a reversed logarithmic scale. Logarithmic because a difference of 1 in this scale corresponds to a brightness factor of  $\sqrt[5]{100} = 2.512$ , and reversed because the brightest the object, the lowest the magnitude. For example, the star Vega has a visual magnitude of +0.03, the Sun has a visual magnitude of  $-26.74$  and a full Moon of  $-12.5$ , being +6.5 the naked eye limiting magnitude (e.g., Vitiello, 2020).

The visual magnitude of debris also relates to their capability to reflect the solar light, which depends on many factors like size, phase angle, reflectivity, and, evidently, the distance from the observer. To calculate the visual magnitude, the debris's absolute magnitude ( $M$ )—not confused as the stellar absolute magnitude—has also been considered (e.g., Clemens, 2019).

$$M = m_{sun} - 2.5 \log\left(p \frac{A}{a^2}\right) \quad (2.1)$$

In equation 2.1,  $m_{sun}$  is the visual magnitude of the Sun,  $p$  is the reflectivity (albedo),  $A$  is the cross-section area, and  $a$  is the distance between the Sun and the Earth in kilometers. Clemens used the debris radius instead of the area, but Hostetler and Cowardin utilized the area (Hostetler & Cowardin, 2019).

Albedo is the ratio between the reflection of the object and the incident energies. It has values from 0, meaning that the object absorbs all the energy, to 1, meaning that all the light is reflected. The latest estimation for the reflectivity of debris is 0.175. The following equation represents the calculation of the debris's visual magnitude ( $m_{debris}$ ).

$$m_{debris} = M + 5 \log\left(\frac{R\Delta}{a^2}\right) - 2.5 \log(F(\phi)) \quad (2.2)$$

In this equation,  $R$  represents the distance from the debris to the Earth,  $\Delta$  is the distance from the debris to the Sun, and  $(F(\phi))$  is the solar phase angle function. Equation 2.2 is actually a general formula to calculate the visual magnitude of any object reflecting the Sun's light for an observer on Earth (see, Clemens, 2019).

The phase function depends on the solar phase angle ( $\phi$ ), a crucial parameter to measure the visual magnitude. This angle brings the information of the illumination condition of an object in space, and it is defined as the Sun-object-observer angle (Figure 6). Better illumination conditions are obtained for smaller phase angles. As this angle increases, the bright side of the object starts facing away from the observer. The best scenario is obtained when  $\phi = 0^\circ$ , and the worst when  $\phi = 180^\circ$ , but different reflecting surfaces may have different phase functions (Vitiello, 2020).

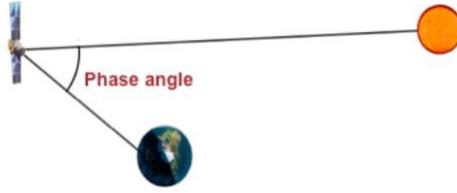


Figure 6 Phase angle: Sun-object-observer angle (Kervin et al., 2010)

In order to simplify the calculations, the debris is assumed to be spherical. The solar phase angle function can be split into specular and diffuse components. The specular means that the object's surface is perfectly smooth and reflects the light at the same angle as it was received. The diffuse implies that the surface is not perfectly smooth, so the reflection is not in a single direction (see, Miguel, 2005).

$$F_{spec}(\phi) = \frac{1}{4\pi} \quad (2.3)$$

$$F_{diff}(\phi) = \frac{2}{3\pi^2} [(\pi - \phi) \cos \phi + \sin \phi] \quad (2.4)$$

Equations 2.3 and 2.4 show the specular and diffuse components of the solar phase angle function. These equations are related to the cross-section area. That's why the area was chosen to be used in equation 2.1.

Clemens suggests using a combination of these two components to obtain better results for the visual magnitude. So, a mixing coefficient  $\beta$  is added in equation 2.2 to combine the specular and diffuse components.  $\beta$  ranges from 0 to 1. When equal to 1, it means that the debris is entirely diffuse while 0 is completely specular.

$$m_{debris} = M + 5 \log \left( \frac{R\Delta}{a^2} \right) - 2.5 \log [\beta F_{diff}(\phi) + (1 - \beta)] F_{spec}(\phi) \quad (2.5)$$

## 2.5 Algorithms to detect space debris

Optical space debris detection consists of detecting faint and small moving objects over a fixed background of stars. Space debris observation in *LEO*, given their fast orbital velocity, has a short pass duration and short field of view cross-time. Therefore, these moving objects appear in the image as an elongated spot or broad line, usually know as streaks (e.g., Vananti et al., 2015).

There are several methods and techniques to detect debris. However, they all rely on the same principle: to remove from the image all of the non-streak patterns, like stars and noise, leaving behind only possible sources of space debris.

Xi et al. (2016) propose a methodology using a time-index image produced by recording the time index, an approach to identify the temporal coordinates of specific data. The time index image is generated by recording the time index of the maximum values for every pixel in a frameset of an image sequence.

The algorithm works as follow: first, the image preprocessing is executed to remove the non-uniform background and calibrate the hot pixel and flicker noise, a form of electronic noise that dominates at low frequencies. After, the time-index filtering is used to remove the stars and noise. The debris, stars, and noise differences are very significant in the image by applying this filtering. Debris is a connected region with the same time index, while stars are regions with several time indices, and noise has a random isolated variable time index, as shown in Figure 7. After these two steps, only debris and a few false alarms are left in the image.

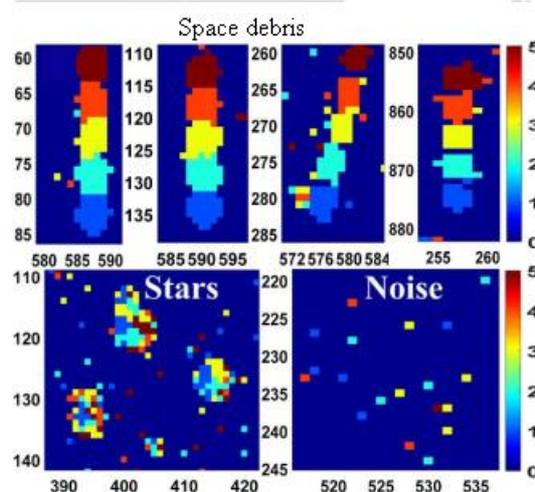


Figure 7 Debris, stars, and noise in the time- index image (Xi et al., 2016)

Vananti et al. (2015) suggest a distinct approach. Their algorithm utilizes image convolution with a spatial filter with the geometric form of the possible streak (see Figure

8). The algorithm's purpose is to generate a more or less perfect match between the streak and the filter, changing the filter's orientation and length. Before applying the filter, the background is subtracted from the background intensity distribution based on a Gaussian fitting, and the stars are removed up to a certain cut-off intensity.

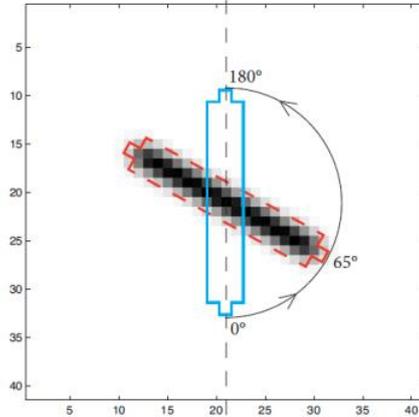
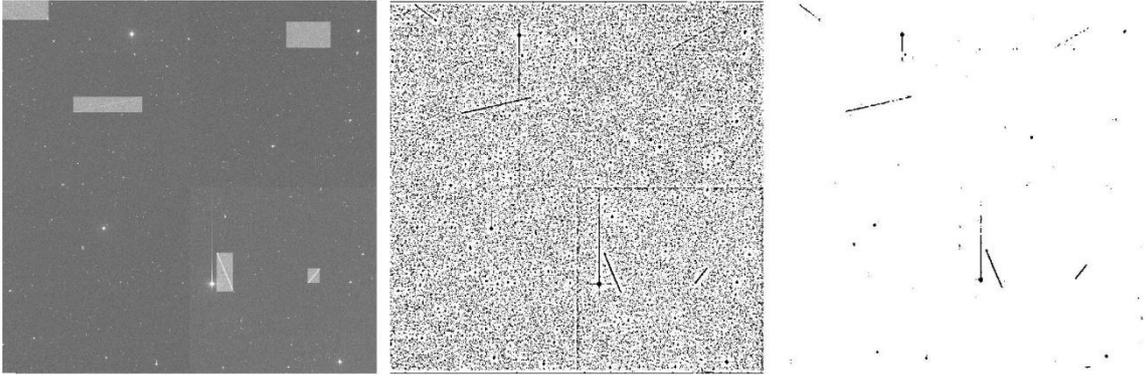


Figure 8 Example of a perfect match with the filter proposed (Vananti et al., 2015)

Hickson (2018) describes yet another method that combines matched filtering with a fast discrete Randon transform to reach high sensitivity and speed. The Randon transform, which is defined as the integral along all the lines, decreases the number of search dimensions. Applying this transform, detecting long streaks becomes a problem to detect local maximum, improving speed. Once again, before using the method, the image is processed to mask the stars and noise, and also the background is subtracted. After this, the Randon transform is computed to find the highest values, and the matched filtering is used to detect possible streaks.

Besides the detection of space debris, some methodologies propose a more complex study. For example, that is the case of the ESA-funded StreakDet (Virtanen et al., 2016). This algorithm consists of three procedures: segmentation, the first part of the method in which all the necessary information is extracted from the image; a classification that is the characterization of the extracted data; and astrometric and photometric reduction, tools for coordinate and magnitude conversion.

The segmentation is the most interesting part of this method for this work. It is applied to separate unknown transients objects from an image with many known static objects (stars). First, a grayscale mean-filtering operation is done to emphasize the difference between the background and the image features. After the stars, background, and noise are removed from the grayscale image resulting in an image with streaks only. This procedure can be seen in Figure 9.



*Figure 9 Segmentation process (Virtanen et al., 2016)*

Another procedure is proposed by Kim (2016). The author developed an algorithm known as ASTRiDE (Automated Streak Detection for Astronomical Images) to detect streaks using each object's border and morphological parameter. The border is identified applying the Boundary tracing method, a segmentation technique that recognizes the boundary pixels of the digital image. ASTRiDE also employs an image subtraction to reduce the confusion caused by the dense distribution of faint stars. Then, ASTRiDE quantifies the shape of each border to determine if it is a streak. Thus the algorithm can detect long, short, and curved streaks.

### 3 Star Tracker

A star tracker is an attitude sensor present in a spacecraft. Its primary purpose is to provide attitude information to the spacecraft's guidance, navigation, and control system. In these objects, stars are extracted by an image taken by the star tracker's sensor. The detected stars are compared to the store catalog to determine the spacecraft's orientation (Pedrotty et al., 2020). Star trackers are the most accurate attitude sensors and can also detect flying objects due to their reflectivity, such as space debris (Ettouati et al., 2006).

#### 3.1 Architecture and operation of a star tracker

A star tracker comprises an optical system, image sensor, and electronics and signal processing system. The optical system is responsible for focusing the starlight into the image sensor, wherein the information of the sky is extracted in image format and transferred to the electronics and signal processing system. This last system is responsible for processing the image and determination of the attitude (Lopes, 2018)

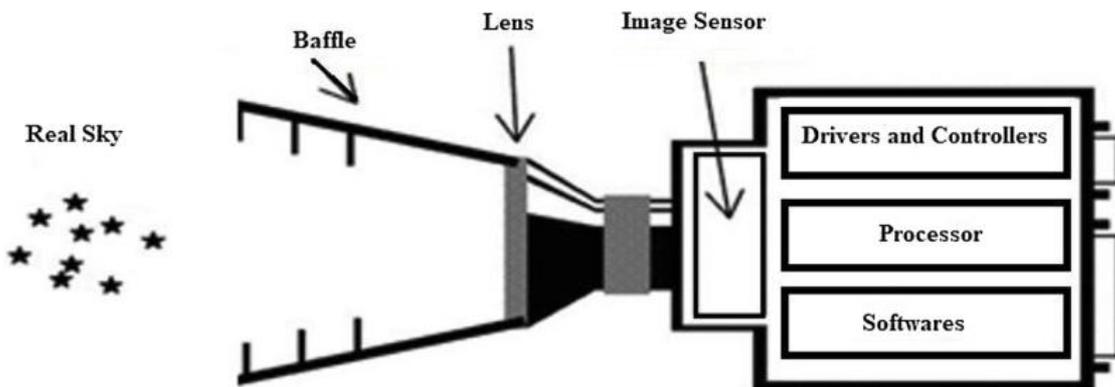


Figure 10 Example of a typical star tracker. Adapted from (Sarvi et al., 2020)

The baffle and the set of lenses are part of the optical system. The baffler is a light shield that protects the star tracker from light that affects its functioning, like the Sunlight. The image sensor can be a CCD (Charge Couple Device) or a CMOS (Complementary metal-oxide-semiconductor). The electronics and signal processing system is composed of drivers, controllers, processors, and software.

The operation of the star tracker is the following: first, the image taken by the sensor is processed to detect any stars, then the centroid of the existing star is extracted. After, the selected algorithm uses a given feature to identify the star by searching in the database and matching the captured stars to the stars in the catalog. Then the match of the

orientation of the captured stars to the star catalog is done by matrices transformations that illustrate the satellite position (Sarvi et al., 2020).

### **3.2 Image sensor**

The image sensor is responsible for forming the image of the objects observed by the optical system. This sensor can be CCD or CMOS. Each one has its advantages, and it is chosen depending on the mission specification.

CCD is a silicon chip sensitive to light divided into many independent pieces known as pixels. CCD is a device that measures how much light falls on each pixel. The output is a digital image consisting of a matrix of numbers, where each number corresponds to one pixel and represents the amount of light that falls on that pixel (e.g., Romanishin, 2002). The operation of the CCD is the following: incoming photons collide with the silicon within a pixel. They are absorbed if they have sufficient energy to interact with the CCD material (silicon). Silicon is chosen because it can easily absorb the energy in the visible range. This photon absorbed makes the valence electrons move to the conduction band. Once the electrons have been freed on the conduction band, they are collected and held until the readout occurs (Howell, 2006).

A CCD structure consists of an array of electrodes running orthogonally to a set of isolated charge transfer channels. Biasing of the electrodes phases produces an array of pixels that gather the photon generated electrons. At a given exposure, the imaging area electrodes are pulsed or clocked to transfer the charge of the integrated image down the array one line at a time. The lowest line is transferred into a serial readout register, which runs in an orthogonal way to the image area transfer channels or columns. This register allows that each pixel is read out sequentially through a charge detection amplifier. Once the complete line has been read, the imaging area electrodes are clocked again for the following image line to be transferred into the readout register. This sequence is repeated until the CCD reads all the image lines (Waltham, 2013). This procedure is illustrated in Figure 11.

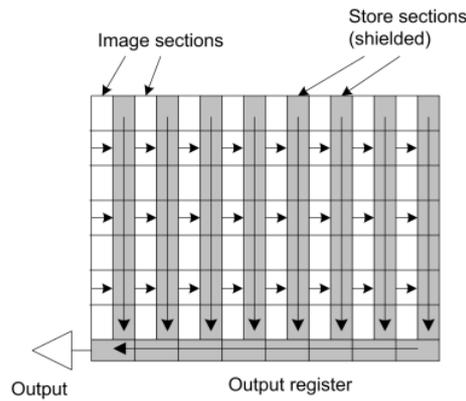


Figure 11 Structure of a CCD showing that the image is read line by line. Adapted from (Waltham, 2013)

CMOS is a semiconductor device that explores the same silicon chip technology used in microprocessor systems, in which several transistors can be integrated on a single silicon circuit. Thus, this provides the opportunity to embed an extensive array of pixels, each with its photodiode and readout transistors, alongside all the necessary electronics needed to address the array (Waltham, 2013).

The simplest CMOS pixel consists of a photodetector and three transistors. The photodetector detects the light energy, and each transistor has a different function. One pre-charges the photodetector, the other senses the signal voltage on the photodetector, and the last one selects the row. The pixel array is accessed one row at a time. At the bottom of the array, the individual pixels within the row are picked and read out columns-by-columns. Thus, contrary to the CCD, where the charge is stored and then read sequentially, each pixel is an individual readout circuit in CMOS (Waltham, 2013).

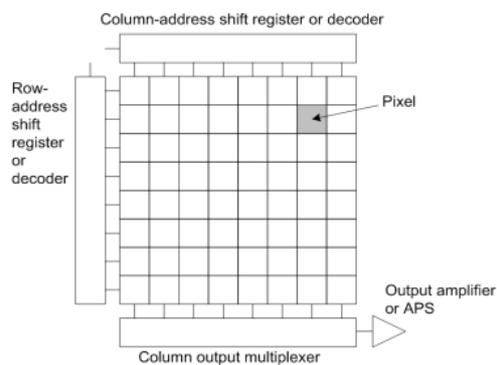


Figure 12 CMOS array of pixels. Adapted from (Waltham, 2013)

So, both types of sensors are composed of a grid of pixels. CMOS has the advantages of fast readout, low power consumption, and low cost, while CCD is more sensitive to light and tends to have lower noise (Jain, 2016).

### 3.2.1 Astronomical photometry

In astronomy, photometry is the science to measure the flux received by a celestial object. In the form of electromagnetic radiation, the flux is the amount of energy incoming from the referred object (Romanishin, 2002). This information is used to calculate the magnitude of stars or other celestial objects.

As said in chapter 2, the magnitude is a quantity measured in a logarithmic inversed scale that describes how bright an object is. The magnitude ( $m$ ) is calculated by comparing a standard star and the star or celestial object of interest.

$$m - m_0 = -2.5 \log\left(\frac{\Phi}{\Phi_0}\right) \quad (3.1)$$

The index 0 represents the standard star, and  $\Phi$  is the flux. Equation 3.1 can be rearranged to give the flux if the magnitude is known.

$$\Phi = \Phi_0 10^{-\frac{m - m_0}{2.5}} \quad (3.2)$$

These magnitudes are known as apparent magnitude because they are related to the flux of a standard star, and it is defined as the apparent brightness of a star or celestial object when observed on the Earth. There is also a widely used distinct type of magnitudes, the (stellar) absolute magnitude, which is the star's intrinsic brightness when observed 10 parsecs (1 parsec  $\cong$  3.26 light – years) away from the Earth, and the absolute magnitude for solar system objects. The last one is related to the object's brightness when being simultaneously at 1 AU from the Sun and 1 AU from the Earth with a phase angle of 0 deg (Lopes, 2018).

The standard stars are A-type main-sequence stars with a magnitude equal to zero. These types of stars are the most brilliant stars in the night sky to the human eye. On Earth, the star with magnitude zero corresponds to the Vega star (Lopes, 2018).

The electromagnetic radiation covering a range of wavelength of at least  $10^{16}$  from the high photon energy gamma rays with a short wavelength to low energy radio photons with a long wavelength (Romanishin, 2002).

When the photons emitted from the stars reach the sensor, they are converted into electrons. The longer the integration time (interval during which the sensor is retained charge), the more energy the sensor received and the higher the number of electrons. The energy of a photon is given by

$$E_\gamma = \frac{hc}{\lambda} \quad (3.3)$$

where  $h$  is the constant of Planck,  $c$  is the speed of light, and  $\lambda$  is the wavelength. Equation 3.3 can transform the flux coming from the star in a given bandwidth of photons. The amount of photons collected by the sensor from the object of interest is known as signal (Howell, 2006).

The signal is affected by the f-number of the lens because it represents the photons' number that crosses the lens. The f-number is the ratio between the focal ( $f$ ) lens and the diameter of the system's aperture ( $d$ ), Equation 3.4. Higher the f-number, lower the aperture, which means fewer photons arrive into the image sensor (Lopes, 2018).

$$f\# = \frac{f}{d} \quad (3.4)$$

### 3.2.2 Signal

The incoming photons from the stars are transformed into photoelectrons by the image sensor, composed of a grid of pixels. The imaging formation starts when the light reaches the array and spreads out in a few groups of pixels. Then, the convolution product between the object and the image system's point spread function ( $PSF$ ) produces the image (Lopes, 2018).

$PSF$  describes the impulse response of a focused optical system to a point source. The shape of the  $PSF$  depends on which factors are limiting the image quality. If imperfections in the lenses do not modify the image quality, the diffraction is the only limiting factor, so the minor point that a lens or a mirror can focus on a beam of light is the size of the Airy disk. So then, the image of a star observed by a perfect lens is expected to be an Airy diffraction disk with a Gaussian-like form.

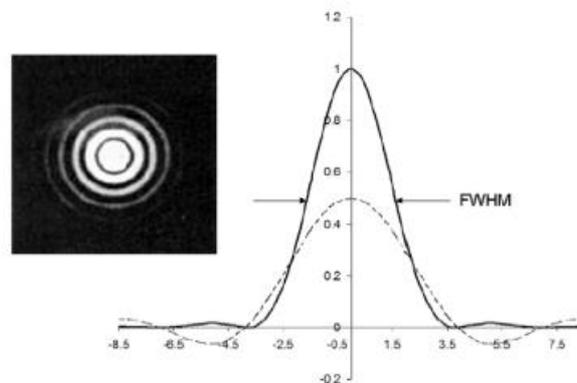


Figure 13 Point spread function of a star with the format of an Airy diffraction disk (Lopes, 2018)

The diameter of the Airy disk can be determined knowing the diameter and the f-number (f#) of the lens, and the wavelength:

$$D_{Airy} = 1.22\lambda \frac{f\#}{D_{lens}} \quad (3.5)$$

Besides the convolution between the object and the image system's point spread function, the noise and the star magnitude are also essential factors in image formation. The noise is a negative factor because it affects the image sensor's efficiency, and brighter stars have a more intense distribution than dimmers. Thus, the sensors have a magnitude limit to detect stars since the sensor's noise often hides the light distribution of dimmer stars (Lopes, 2018).

### 3.2.3 Noise analyses

The noise describes the uncertainty of the measurements due to the electronics fluctuations in the equipment. In sensor images, the source of noise can be divided into three regimes: readout noise ( $N_{RN}$ ), photon shot noise ( $N_{SN}$ ), and dark noise ( $N_{DN}$ ) (Howell, 2006). The total noise ( $N$ ) can be calculated as:

$$N = \sqrt{N_{RN}^2 + N_{SN}^2 + N_{DN}^2} \quad (3.6)$$

The readout noise is related to the electronics of the sensor. The dark noise and the shot noise are proportional to the sensor's integration time. Higher values of integration time lead to a higher number of photons absorbed, but this increases the noise (Lopes, 2018). The noise can be quantified by image acquisition. However, to do so, the charge conversion efficiency ( $CCE$ ) has to be determined. The  $CCE$  is a quantity that defines the relation between the number of electrons converted and the resulting output in quantities. The  $CCE$  enables the determination of the Quantum efficiency ( $QE$ ).  $QE$  represents the ratio between incident photon and converted electron, allowing an evaluation of the relative collection power of the device as a function of the wavelength (Howell, 2006). Readout noise, or just read noise, is the number of electrons introduced per pixel into the final signal of the device due to the noise generated by all the electronics (Howell, 2006). This value depends on the sensor's  $CCE$  and can be measured by extracting pictures in the dark with the integration time near zero.

$$N_{RN} = \frac{CCE \sigma}{\sqrt{2}} \quad (3.7)$$

In equation 3.6,  $\sigma$  means the standard deviation from the pixel signal of the picture in the dark. Therefore, the read noise can not be controlled because it only depends on the quality of the sensor's electronics (Lopes, 2018).

Shot noise describes the fluctuation of photons ( $\gamma$ ) detected and converted to photoelectrons by the image sensor. It can be calculated as:

$$N_{SN} = \sqrt{\gamma} \quad (3.8)$$

Equation 3.8 shows that the shot noise depends on the exposure time since the number of detected photons increases with the exposure time (Lopes, 2018).

The dark noise is the number of electrons thermally generated by the sensor. It depends on the integration time ( $t$ ) and the dark current ( $i_{DC}$ ). According to Howell (2006), the dark current is specified as the number of thermal electrons generated while the sensor operates in the dark.

$$N_{DN} = \sqrt{i_{DC}t} \quad (3.9)$$

So, the dark noise increases with temperature. Therefore, for lower values of dark noise, the device should have a thermal cooling system to decrease the temperature of the electronics components (Lopes, 2018).

Another crucial concept in terms of the image sensor is the signal-to-noise ratio ( $SNR$ ), a quantity that indicates the accuracy of the measurements by comparing the amount of signal received with the background noise level.  $SNR$  is often given in decibels ( $dB$ ), and if its value is positive, the signal overlaps the noise, and the sensor can read it (Lopes, 2018).

$$SNR = 10 \log\left(\frac{signal}{N}\right) \quad (3.10)$$

### 3.3 Star detection algorithms

Any star tracker utilizes a given algorithm to identify the stars and extract the attitude. These algorithms have two modes of operation: initial attitude acquisition and tracking mode. The initial attitude acquisition, or lost-in-space algorithm, does not need any initial information to determine the attitude. On the other hand, the tracking mode requires some information to work correctly. However, the star trackers operate with these two modes. In the beginning, the lost-in-space method extracts the attitude by looking at a group of stars. Then, the star tracker switches to the tracking mode and follows the same set of stars identified by the first mode. If this information is lost, the star tracker goes back to the initial attitude acquisition and repeats the same process (Spratling & Mortari, 2009). In general, these techniques use inter-star angles, the angle between two stars from the camera's perspective, the brightness of the stars, and some computation of these values

to identify the stars (Spratling & Mortari, 2009). In this work, it was studied only the lost-in-space algorithm.

There are several studies related to lost-in-space algorithms. After a survey, three open-source algorithms were selected: Tetra (Brown et al., 2017), OpenStartracker (Tennenbaum, 2017), and SOST (Gutiérrez et al., 2020).

### **Tetra**

Tetra is a lost-in-space star identification algorithm that utilizes the minimum possible computation time and number of database accesses, in most cases precisely one. The authors claim that an asymptotically faster algorithm than Tetra can not exist (Brown et al., 2017).

Tetra needs three or four stars to work, but it can operate on two-star patterns if information that disambiguates the pair of stars is available, star brightness, for example. However, the recommended number of stars is three or four. Tetra identifies four neighbor stars and divided the distance between them by the brighter star from this group of stars. This information is stored in array format, and then, each array is compared to the intern catalog to match the stars and give the attitude information (Brown et al., 2017).

### **OpenStartracker**

OpenStartracker is a framework in which, given a set of sample star images, it can calculate all the parameters, generate the star tracker database, and perform testing and validation. The framework developed consists of three components: a high-speed star tracker algorithm; an optimal star tracker database generation; and an automatic test that validates the star tracker under various conditions (Tennenbaum, 2017).

The high-speed star tracker algorithm operates by the following: first, a background image is generated to subtract the average noise background and dead pixels. Then the image parameters are solved using Astrometry.net (Lang et al., 2010), and the orientation and the field of view of the images are determined. After that, the centroid technique, a method to calculate the centroid of a group of stars using the brightest one as a reference, is used to extract the position of the stars (Tennenbaum, 2017).

The database is generated using the stars' information, save the star id's and the distances between them as a binary k-vector file. Finally, the framework is validated using the automatic test to determine the expected slew tolerance of the star tracker under various conditions (Tennenbaum, 2017).

## **SOST**

SPEL (Space and Planetary Exploration Laboratory at the University of Chile) Open Star Tracker (SOST) is an ultra-low-cost solution to estimate the attitude in the lost-in-space scenario. It is an algorithm developed to work in a Raspberry Pi (Gutiérrez et al., 2020). The attitude estimation process starts with the extraction of the stars in an image using the Source Extractor, a software utilized to detect astronomical sources. Besides detecting stars, this software is also applied to calculate the star's position and brightness magnitude. After that, the list of objects from the image is compared with a star catalog using the Match, a software that establishes a relationship between two different lists. Once the catalog and the image sources are paired, the attitude is determined (Gutiérrez et al., 2020).

## **Astrometry.net**

Astrometry.net is a lost-in-space algorithm that takes as input an astronomical image and returns as output the scale and the orientation of that image. It can be used on the website *nova.astrometry.net*. This algorithm first identifies the astronomic sources in the image by running some image-processing steps. Next, the system divides the stars into a subset of four stars and determines the relative position of each star inside a given subset. Then, the algorithm searches, in a pre-computed catalog with relative positions, almost identical matches between the catalog and the image. Each match provides information about the location, scale, and orientation of the image in the sky (Lang et al., 2010).

Astrometry.net appears to be the perfect lost-in-space algorithm based only on the extraction of stars in the image and the match with the catalog. However, it often takes tens of seconds to run and requires large amounts of memory to store the database (Brown et al., 2017). Therefore, this algorithm is used as a tool to compare and validate the results among others algorithms.

### **3.4 Market survey**

Star trackers are devices with a particular technology that are developed to satisfy the mission requirements. So, the properties of star trackers are not standard. That is why their costs can vary in a range between US\$ 30,000 to US\$ 140,000 (Clemens, 2019).

Table 2 shows the limiting magnitude and field of view (FOV) of some star trackers. These results were necessary to define optimal values for the system developed in this work.

Table 2 Survey of star trackers

<i>Manufacturer</i>	<i>Model</i>	<i>Limiting magnitude</i>	<i>FOV (deg)</i>
<i>Blue Canyon Technology</i>	Standard NST	7.5	10 x 12
<i>Ball Aerospace</i>	HAST	5.5	9.47 x 9.47
<i>Jena Optronik</i>	ASTRO 15	6.5	13.25 x 13.25
<i>Berlin Space Technologies</i>	ST200	6	
<i>OCE Technology</i>	PST3	5.8	15 x 12
<i>Sattelite Services Ltda</i>	Star Tracker	6	18.6 x 12

## 4 Optical system

The selection of the optical system is a crucial task for the star tracker. So, this chapter reviews the sensor's characterization done by Lopes (2018) and also it is calculated the FOV and the limiting magnitude of the system.

### 4.1 Image sensor

The image sensor chose for this work was the Apatina MT9J001, a monochromatic, 10 megapixel CMOS. According to the datasheet, the sensor is described as a breakthrough low-noise CMOS imaging technology that achieves near CCD image quality, keeping CMOS's inherent size, cost, and integration advantages. Some characteristics of the sensor are shown below.

Table 3 Characteristics of the MT9J001 10 Mp CMOS digital image sensor

<i>MT9J001</i>	
<i>Active image size</i>	6.440 mm (H) x 4.616 mm (V)
<i>Active pixels</i>	3856 (H) x 2764 (V)
<i>Pixel size</i>	1.67 x 1.67 $\mu\text{m}$
<i>SNR<sub>MAX</sub></i>	34 dB

The MT9J001 camera header board is connected with the Arducam USB2 Camera Shield. Arducam is a general USB camera control board developed for both computers with Windows or Linux and embedded systems like Raspberry Pi. The MT9J001 is connected with Arducam using a 30pin ribbon cable in the secondary camera interface.

The sensor is controlled by codes available in the Arducam USB Camera Shield GitHub. It is possible to find the codes in Python or C++. In Windows, the code is also found in an interface user software called USBTest, which allows the user to control the camera register and specification for integration time.



Figure 14 MT9J001 connected with the Arducam USB2 Camera Shield through a 30pin ribbon cable

### Integration time

According to the sensor datasheet, the integration time can be calculated as

$$\begin{aligned} & \textit{integration\_time} && (4.1) \\ & = \frac{(\textit{coarse\_integration\_time} \times \textit{line\_length\_pck}) + \textit{fine\_integration\_time}}{\textit{vt\_pix\_clk\_freq\_mhz} \times 10^6} \end{aligned}$$

The values of *line\_length\_pck*, *fine\_integration\_time* and *vt\_pix\_clk\_freq\_mhz* are fixed to facilitate the calculations. The *line\_length\_pck* is related to the number of pixel clocks read per row, and it is equal to 7440. *vt\_pix\_clk\_freq\_mhz* is the pixel clock for the image sensor. According to the Arducam datasheet for the MT9J001, the pixel clock is 24 MHz. The value of *fine\_integration\_time* is the minimum allowed by the sensor, 1010.

The *coarse\_integration\_time* can be controlled by the register values available in the sensor datasheet. For this work, the register value used is 12306 ( $0 \times 3012$ ). With this register, a value of 50 for the *coarse\_integration\_time* means an integration time of 15.5 ms.

## 4.2 Sensor characterization

The optical system chosen is the same one used by Lopes (2018). The sensor was available for the application in this work. As said in chapter 3, star trackers are expensive, so Lopes developed a low-cost star tracker. For this, the author already did the characterization of the sensor.

### 4.2.1 Charge conversion efficiency

As said in chapter 3, the charge conversion efficiency (CCE) relates to the number of electrons converted and the digital counts. According to Lopes, CCE can be measured by focusing a monochromatic and homogeneous source of light in the sensor.

To determine CCE, Lopes did a plot of the variance and the mean counts of images extracted for the source of light in different integration times until the sensor reaches saturation, which is the maximum value of a pixel. For this, the author utilized an integrating sphere and a Neon-Helion gas laser beam as a source of light.

The laser beam guarantees a light source monochromatic, and the integration sphere ensures that the light reaches the sensor homogeneously. It also acts as a baffler, preventing that external light interferes with the measurements. The integration sphere has three cavities where any instrument can be mounted. Lopes put the laser in one cavity, the sensor in the other, and close the remaining one, as shown in Figure 15.

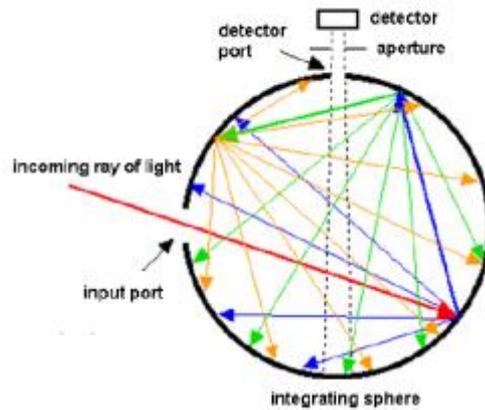


Figure 15 Integration sphere and experimental setup did by Lopes (Lopes, 2018)

After that, it was taken ten frames varying to the minimal integration time,  $6.31 \mu s$  according to Lopes, to the value in which the sensor becomes completely saturated,  $71.43 ms$ . The mean counts and the variance were extracted for each frame. However, the graph of the mean count versus the variance is not linear. For low levels of light, the total noise is dominated by the readout noise, and for higher levels, the total noise is dominated by the shot noise.

So, the author chose the region of the graph where the mean count and the variance have an approximate linear behavior. Then a linear line was fitted in these data, and the CCE corresponds to the slope of the linear fit. The value of CCE found is  $5.22 e^- / ADU$ , where ADU is Analog-to-Digital Unit.

#### 4.2.2 Readout noise

The readout noise can be measure by a bias frame that is an image extracted with the shutter close in a zero integration time. However, Lopes affirms that it is impossible to achieve zero integration time, so for the determination of the read noise, it was utilized the minimal value found for this quantity:  $6.31 \mu s$ .

The author mounted the sensor with the shutter close in the integration sphere to measure the readout noise. The other two cavities were closed, and the lights of the laboratory were turned off. After that, two frames were extracted with the minimal value for the integration time. The frames were subtracted by each other, resulting in the structure of the noise.

Lopes found the value of  $\sigma = 3.24 ADU$  for the standard deviation of the resulting frame. Substituting the results of  $\sigma$  and CCE into equation 3.6 gives the output for the readout noise, which is  $11.97 e^-$ .

### 4.2.3 Dark noise

The dark noise is calculated by the dark current, which is measured by taking dark frames with a thermal control system. A dark frame is an image extracted in the dark with the shutter closed. The difference between dark and bias frames is related to the integration time. The first one is obtained using integration time different from zero.

Lopes (2018) decided not to control the temperature, and the measurements were done in environmental temperature, which was 24 °C.

The measurements were done in the integration sphere with the same configuration of the readout noise. Thus, ten dark frames were taken for different integration times until the sensor reached a reasonable saturated pixel level (hot pixels). After that, a mean of the ten dark frames for each integration time was calculated. With this value, a plot of mean counts vs integration time was created.

So, Lopes did a linear fit in these points. The slope of the line represents the dark current, which has a value of  $1.095 e^-/s$ . Substituting into equation 3.9 with a given integration time, then it is possible to obtain the dark noise.

### 4.3 Optical system FOV

There were used two optical systems in this work—the image sensor with a lens and in a telescope.

The lens chosen was the same that Lopes (2018) utilized in her star tracker. The lens is from a Chinese brand, Fujian, and it has an f-number of 1.6 and a focal length of 35 mm. There is no datasheet for this lens, so the transmission coefficient was predicted by the author.



*Figure 16 Optical system: lens + sensor*

The FOV of this optical system can be calculated by

$$FOV = 2 \tan^{-1} \left( \frac{h_{sensor}/2}{f} \right) \quad (4.2)$$

The  $h_{sensor}$  is the active image size of the sensor and  $f$  is the focal length. So the FOV is 10.5 *deg* horizontally and 7.5 *deg* in the vertical.

The optical system, sensor and telescope, was used because it is easier to point to a specific region of the sky. The telescope is a Celestron EDGE HD 14 inches. It has a focal length of 3910 mm. This is the telescope of the OGAUC - Geophysical and Astronomical Observatory of the University of Coimbra.



*Figure 17 Optical system: sensor + telescope*

The FOV, using equation 4.2, is 5.6 *arcminutes* horizontally and 4 *arcminutes* in the vertical.

The FOV of the lens is much larger than the telescope. However, the telescope allows controlling where the optical system is pointing at the sky.

#### **4.4 Prediction of the limiting magnitude**

The prediction was made only to the optical system: sensor and lens. The limiting magnitude is the threshold value that the image sensor can detect for magnitude. However, it isn't easy to calibrate astronomical photometric equipment to measure the magnitude because the light is not monochromatic and the atmosphere's variability. So, this measurement is done in terms of the brightness of sources in specific wavebands relative to standard stars. For example, the Johnson-Cousins system is a method developed to measure the apparent magnitude of stars in the ultraviolet (U), visible (V), blue (B), red (R), and infrared (I) wavebands (Lopes, 2018).

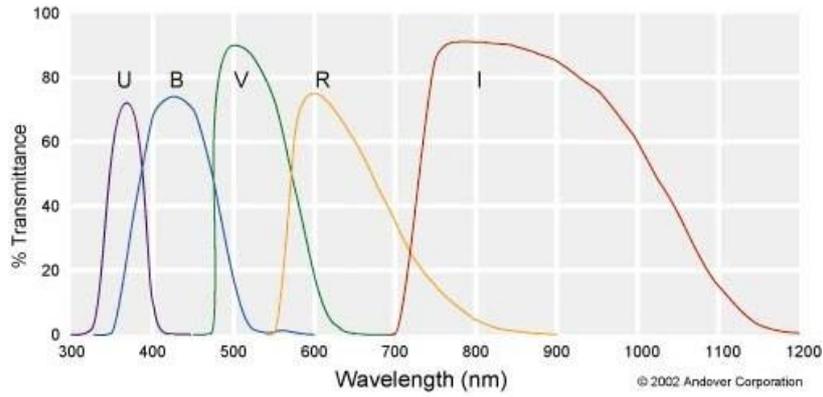


Figure 18 Johnson-Cousins system filter curves. Adapted from (Lopes, 2018)

As said in chapter 3, the standard stars are A-type main-sequence stars, A0 V, which are the most brilliant stars in the night sky to the human eye. On Earth, the star with magnitude zero corresponds to the Vega star. The flux calibration values to these types of stars in the Johnson-Cousins system are in Table 4.

Table 4 Flux calibration values of A0 stars for the Johnson-Cousins system. Adapted from (Lopes, 2018)

Waveband	$\Phi_x [W m^{-2} \text{\AA}^{-1}]$	$FWHM [\text{\AA}]$	$\lambda_x [nm]$
<i>U</i>	$4.22 \times 10^{-12}$	600	360
<i>B</i>	$6.40 \times 10^{-12}$	940	440
<i>V</i>	$3.75 \times 10^{-12}$	880	550
<i>R</i>	$1.75 \times 10^{-12}$	1380	710
<i>I</i>	$8.40 \times 10^{-13}$	1490	970

In Table 4, FWHM is the abbreviation of full width half maximum and  $\lambda_x$  is the wavelength. The total flux of an A0 star is calculated using the flux for the different wavebands ( $\Phi_x$ ).

$$\Phi_0 = \int \Phi_x d\lambda \quad (4.3)$$

So, the integral should cover all bands and take an account the overlap among them. This isn't easy to calculate, and then, sometimes, approximations have to be made. First, by evaluating the spectral sensitivity curve, a graph that provides the quantum efficiency values in different wavelengths, for the MT9J001 available in the datasheet, it can be noted that the sensor covers a range from 400 to 1000 nm.

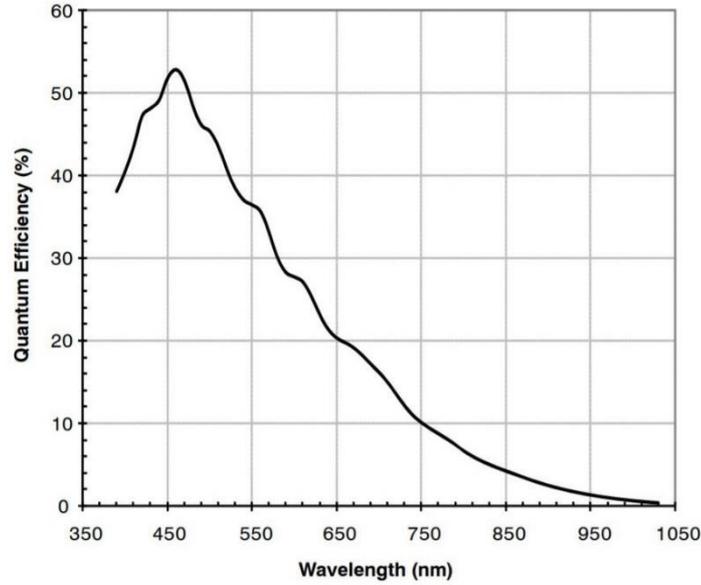


Figure 19 MT9J001 spectral sensitivity curve. Available in the sensor datasheet

This wavelength range means that only the wavebands B, V, R and I have to be considered in the calculations. It was assumed that the wavebands in the Johnson-Cousins system are squares instead of curves, Figure 20, to calculate the flux of the A0 star.

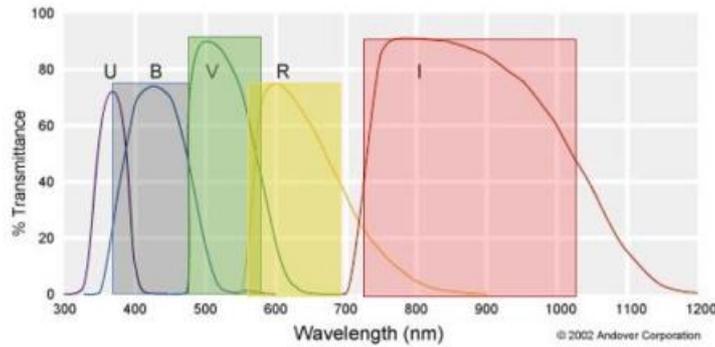


Figure 20 Approximation in the Johnson-Cousins system to measure the flux. Adapted from (Lopes, 2018)

Then, with this approximation, the flux of an A0 star measured by the sensor can be calculated as

$$\Phi_0 = \sum_{x=B,V,R,I} \Phi_x FWHM_x QE_{\lambda_x} \quad (4.4)$$

Equation 4.4 represents the flux in the different wavebands measured by the sensor.  $QE$  is the quantum efficiency of the MT9J001 in the bands. These values can be extracted from Figure 19.

It is expected a higher final result than it supposes to be because the square does not consider the overlap of the wavebands and is not a good representation of the curve. Even so, the value of the flux may be a good estimation.

The flux of the A0 star was calculated by substituting the values for the different wavebands in equation 4.4. After that, equation 3.2 was used to calculate the flux for several values of magnitudes. These results were in units of  $W/m^2$ .

Equation 3.3 was utilized to convert the flux unit for  $photons/m^2$ . This is done by divided each waveband flux by its photon energy. Then the results are multiplied by the lens transmission coefficient, which is 0.7, and by its aperture area. Now the flux has units of  $photons/s$ .

Then, the results were divided by the pixels' number an A0 star is expected to occupy in the sensor and multiplied by the integration time. The outcome is the signal of the sensor for the different magnitudes. According to Lopes (2018), the expected number of pixels for an A0 star is 15 pixels.

The total noise can be calculated by equation 3.6. The square root of the signal gives the shot noise, and the values of readout and dark noise are the ones estimated previously in this chapter.

With values of signal and total noise, it is possible to calculate the signal-to-noise ratio (SNR), equation 3.10. The limiting magnitude is found when  $SNR = 1$ , red line in Figure 21.

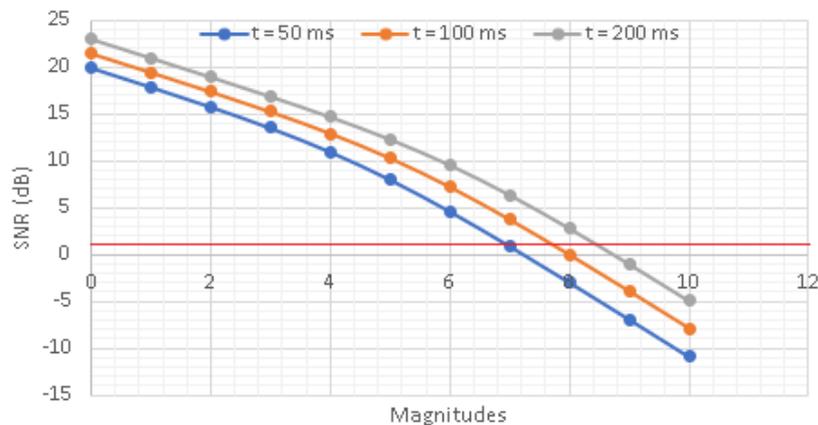


Figure 21 SNR for different magnitudes in distinguished integration time.

According to Figure 19, the limiting magnitude of the MT9J001 with the lens is 7 for an integration time of 50 ms.

### Outcome

The predicted value for the limiting magnitude does not guarantee that the optical system, sensor and lens, will detect objects as fainter as 7 of magnitude due to all the approximations and assumptions made to calculate the flux and noise. Even so, this final result ensures that the camera can detect stars.

## 5 Algorithms

The star detection algorithm is essential in the star tracker operation. Without a good algorithm, the star tracker can not orient itself compromising the mission. Besides that, this work needs an additional algorithm to detect debris. So, this chapter describes the two algorithms that were used.

### 5.1 Tetra algorithm

Among those star detection algorithms of chapter 3, Tetra was chosen for being a robust algorithm in detection, extremely fast, and easy to use. SOST could not detect any of the images tested, and OpenStartracker is very difficult to install and operate. Astrometry.net is not an algorithm developed for star trackers, so it was used to compare the results obtained from Tetra.

In chapter 3, Tetra was described in a concise and summarized way. In detail, after the acquisition of the image, the identification of the stars takes milliseconds. First, the image is converted into a grayscale, and the following parameters are extracted: mean, standard deviation, height, and width of the counts. The counts are read pixel by pixel. The pixels with counts greater than or equal to five times the standard deviation (the user can control this value) are selected for the analysis. The group near the image's borders and the ones with the number of pixels less than five times are excluded. This exclusion process is vital to avoid false detection. The selected groups are presumed to be stars and now are treated as stars. Then, the center of mass for each star is related to its brightest pixel (Lopes, 2018).

After that, Tetra starts by constructing a coordinate system using the four neighbor stars where the largest edge is used to orient and normalize. This system allows each pixel to have coordinates in the image plane. The largest edge is placed along the x-axis, and the coordinates are restricted by two quarter-circles formed by the largest edge constraint, shown in the following figure. Next, Tetra measures the distances among the stars inside the coordinate system, divides the results by the higher one, and stores them in array format. Then, each group has an array with the distance information. Finally, these arrays are compared with the catalog (Brown et al., 2017).

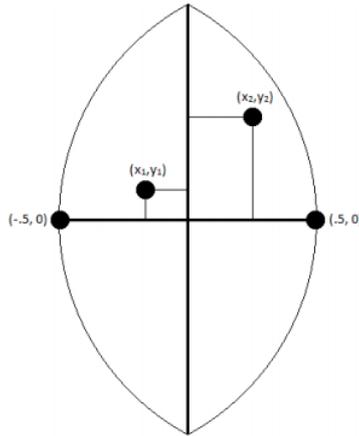


Figure 22 Coordinate system used by Tetra (Brown et al., 2017)

### 5.1.1 Internal catalog

Tetra uses the Yale bright star catalog (BSC5) as an internal catalog. BSC5 has 9110 stars with a limiting magnitude of 6.5. It is stored in hash tables, structures of data implemented in generalizations of arrays. These structures store the information in array format, where each data value has its unique index. (Brown et al., 2017).

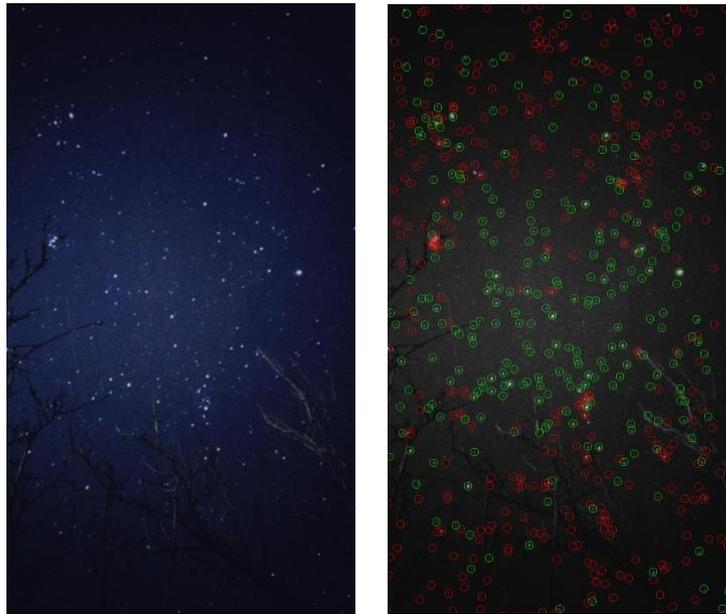
In hash tables, each object has a matching key. In Tetra, the object is the star, and the key is its catalog ID. Each star has an array of information about itself extracted from the catalog to be used by the algorithm. These data are right ascension, declination, magnitude, and catalog ID. The computation time increases with the number of stars in the catalog, so Tetra stores only the stars that the sensor can detect. This limiting magnitude can be defined by the user (Lopes, 2018).

The process of storing the internal catalog into hash tables follows the same principle of detecting stars. Patterns of four neighbor stars are computed, and their distances are divided by the higher one. Each star is stored in the hash tables using its ID as a key. Now inside the hash tables, the matching objects between the catalog and the star detected are their arrays with the normalized distance. The process of building the hash tables only needs to be done once, and it is the task that takes more time: at least one hour.

### 5.1.2 Output

Tetra output is the information about the right ascension, declination, roll, FOV, and mismatch probability (probability of false star detection). Also, the algorithm gives the grayscale image with all the recognized stars inside circles green and red. The stars inside the green circles were identified by the algorithm, while the red ones were not recognized but were considered stars. There are some reasons for these unmatched stars. For

example, they may not be listed in the catalog, or they are fainter than the limiting magnitude defined by the user, and there is also the possibility of not being a star.



*Figure 23 On the left, there is an image of the sky taken from the internet. On the right, it is the same image after being analyzed by Tetra*

### **5.1.3 Change in the internal catalog**

The probability of identifying stars by the algorithm is proportional to the number of stars in the catalog: the larger these values, the greater the star tracker's accuracy. To prove this idea, the preview catalog, BSC5, was replaced by the Hipparcos catalog.

The ESA's astrometric mission, done by the Hipparcos satellite, extracted scientific data from thousands of stars producing the Hipparcos catalog, with 118 218 stars and a limiting magnitude of 12.4 (ESA, 1997). This catalog can be downloaded on VizierR in a file called *I/239/hip\_main*.

The migration to the Hipparcos catalog was done following the work of Lopes (2018). Tetra can read the file *I/239/hip\_main* directly, but the way the algorithm reads the catalog needs some changes. Lopes modified the original code to read 64-bit integer values for star IDs instead of 32-bit because Hipparcos has almost ten times more stars than BSC5 and updated the code to the most recent libraries of Python since Tetra was coded with old version libraries. However, a higher dimension catalog leads to more computational time to produce the hash tables. So the author also created a C code to delete irrelevant information, like blank lines, to decrease some computational time in the process of creating the hash tables.

### 5.1.4 Tests with BSC5 and Hipparcos

Tests using seven pictures of the night sky extracted from several places on the internet were done to evaluate the algorithm's performance with the catalogs BSC5 and Hipparcos. Both were compared, and the values of right ascension and declination were also compared with Astrometry.net.

The pictures were tested by the algorithm using both catalogs. The algorithm for both had the same configuration: limiting magnitude of 6.5, patterns of 5 stars, and the minimum number of 3 pixels to classify the group as a star.

The first difference appeared in the hash tables formations since the BS5 found 9110 stars and Hipparcos 9783 stars. The seven pictures were put in the same folder, and only one run was done to analyze in both cases. The algorithm with BSC5 took 186.97 s to identify all images, and with Hipparcos 203.78 s.

The results obtained for right ascension and declination are shown in the following two tables.

Table 5 Results for right ascension

<i>Right Ascensio [°]</i>	<i>BSC5</i>	<i>Hipparcos</i>	<i>Astrometry.net</i>
<i>Picture 1</i>	242.0067	242.0132	241.428
<i>Picture 2</i>	167.3493	167.356	167.569
<i>Picture 3</i>	284.5506	284.5486	284.581
<i>Picture 4</i>	337.8667	337.8713	337.837
<i>Picture 5</i>	83.3728	83.3656	83.409
<i>Picture 6</i>	61.1307	61.1294	61.146
<i>Picture 7</i>	--	180.9541	180.924

Tetra, with the BSC5 catalog, was not able to identify picture 7. It can be noted that the results between the two catalogs are closer to each other than to those of Astrometry.net. The maximum difference between the values of both catalogs is 0.0072 for picture 5.

Table 6 Results for declination

<i>Declination [°]</i>	<i>BSC5</i>	<i>Hipparcos</i>	<i>Astrometry.net</i>
<i>Picture 1</i>	32.3159	32.3174	32.018
<i>Picture 2</i>	-61.1177	-61.1159	-61.18
<i>Picture 3</i>	36.2307	36.2342	36.232
<i>Picture 4</i>	15.597	15.5989	15.579
<i>Picture 5</i>	4.5408	4.6701	4.66
<i>Picture 6</i>	40.2756	40.2765	40.392
<i>Picture 7</i>	--	-59.2605	-59.264

The same behavior can be noted. The results between the BSC5 and Hipparcos are much closer to each other comparing with Astrometry.net. The maximum difference between the catalogs is 0.1293 for picture 5 again.

Astrometry.net only gives right ascension and declination results, so the values of roll, FOV, and mismatch probability were only compared between the two catalogs.

Table 7 Results for Roll values

<i>Roll [°]</i>	<i>BSC5</i>	<i>Hipparcos</i>
<i>Picture 1</i>	34.0419	34.0378
<i>Picture 2</i>	2.2068	2.2524
<i>Picture 3</i>	104.0805	104.0937
<i>Picture 4</i>	312.583	312.5853
<i>Picture 5</i>	297.4115	297.5896
<i>Picture 6</i>	227.3459	227.3376
<i>Picture 7</i>	--	41.5202

Table 8 Results for FOV

<i>FOV [rad]</i>	<i>BSC5</i>	<i>Hipparcos</i>
<i>Picture 1</i>	56.0801	56.0579
<i>Picture 2</i>	41.5271	41.6264
<i>Picture 3</i>	38.7723	38.7704
<i>Picture 4</i>	51.9076	51.9055
<i>Picture 5</i>	33.2392	33.8964
<i>Picture 6</i>	39.4893	39.4911
<i>Picture 7</i>	--	24.5942

Once again, the maximum differences were observed for picture 5. The maximum roll difference is 0.1781, and for the FOV 0.6572.

Table 9 Results for the mismatch probability

<i>Mismatch probability</i>	<i>BSC5</i>	<i>Hipparcos</i>
<i>Picture 1</i>	$9.15 \times 10^{-136}$	$9.45 \times 10^{-147}$
<i>Picture 2</i>	$9.52 \times 10^{-45}$	$7.36 \times 10^{-40}$
<i>Picture 3</i>	$1.51 \times 10^{-42}$	$1.15 \times 10^{-44}$
<i>Picture 4</i>	$1.55 \times 10^{-45}$	$1.77 \times 10^{-44}$
<i>Picture 5</i>	$1.41 \times 10^{-21}$	$5.83 \times 10^{-42}$
<i>Picture 6</i>	$3.39 \times 10^{-206}$	$4.24 \times 10^{-204}$
<i>Picture 7</i>	--	$1.75 \times 10^{-77}$

It is expected that the larger the number of stars, the lower the value of the mismatch probability. So, Hipparcos should have the smallest results. This behavior can be seen in pictures 1, 3, and 5.

## Outcome

By analyzing the results, BSC5 and Hipparcos had similar performances. BSC5 had three pictures with values closer to the Astrometry.net ones. However, Hipparcos was able to identify an additional image. For this reason, the Hipparcos catalog was chosen to be used in this work.

## 5.2 Debris detection algorithm

In chapter 2 was presented some algorithms with different approaches to detect the streaks. However, the image processing is basically the same: the background and the star are removed, leaving behind only possible sources of debris. These algorithms are not open or are not fully developed. The only one available is the ASTRiDE.

### 5.2.1 ASTRiDE

As said, ASTRiDE is an algorithm based on the outline shape of elongated sources that can detect fast-moving objects with a velocity greater than 10 arcsec/min. ASTRiDE can be found at <https://github.com/dwkim78/ASTRiDE>.

To detect streaks, first, the algorithm extracts the background. This is done by calculating the background level and using its standard deviation to do the removal. After that, ASTRiDE derives the contour map for finding all the borders inside the image. Then, the algorithm uses the morphologies of each border to distinguish streaks from stars. The morphological parameters are:

- Shape factor, which defines the circularity of the source and is calculated as

$$shape\_factor = \frac{4\pi Area}{(perimeter)^2} \quad (5.1)$$

- Radius deviation, an approximated deviation from roundness. It is determined by the standard deviation (*std*) as follow

$$radius\_deviation = \frac{std (distances - radius)}{radius} \quad (5.2)$$

- And area, which is the area inside the identified object border (Kim, 2016).

The shape factor of 1 indicates that the object is a circle, so streaks have a value much smaller than 1. The radius of each object is calculated, and the standard deviation of it is applied to determine the radius deviation. For circles sources, the values of the radius deviation are zero. And, to be considered a streak, the area inside the border must be larger than 10 pixels (Kim, 2016).

The user can control all these parameters. Then, ASTRiDE can be adapt to different scenarios, for example, a more accurate context where the algorithm can detect less evident and shorter streaks or a situation in which it wants to catch only obvious streaks.

### 5.2.2 Output

After the calculations, ASTRiDE generates figures. The image named *all* shows the original image with the found streaks, and it is also created figures zoomed in on each linked streak. The streaks are numbered, and every individual image is denominated relatively to its number.

Besides that, it is also created a *txt* list contained the following information:

- ID, index of the identified streak;
- *x\_center* and *y\_center* that are the coordinates of the streak's center in the image;
- Area;
- Perimeter, number of pixels inside the streak;
- Shape factor;
- Radius deviation;
- Slope, the slope of a linear line fitted to the streak;
- Intercept, the interception of the fitted line;
- And connectivity, an ID of another streak that is probable to be connected to the current streak. If there are no linked streaks, the connectivity value is equal to -1.

### 5.2.3 Tests

Seven pictures extracted from the internet were tested to evaluate the algorithm. These pictures are images of the sky with streaks on them. The source of the streaks is unknown in four of them, and they could be debris or a satellite. In the remaining, the streaks are produced by the STARLINK satellites and the ISS.

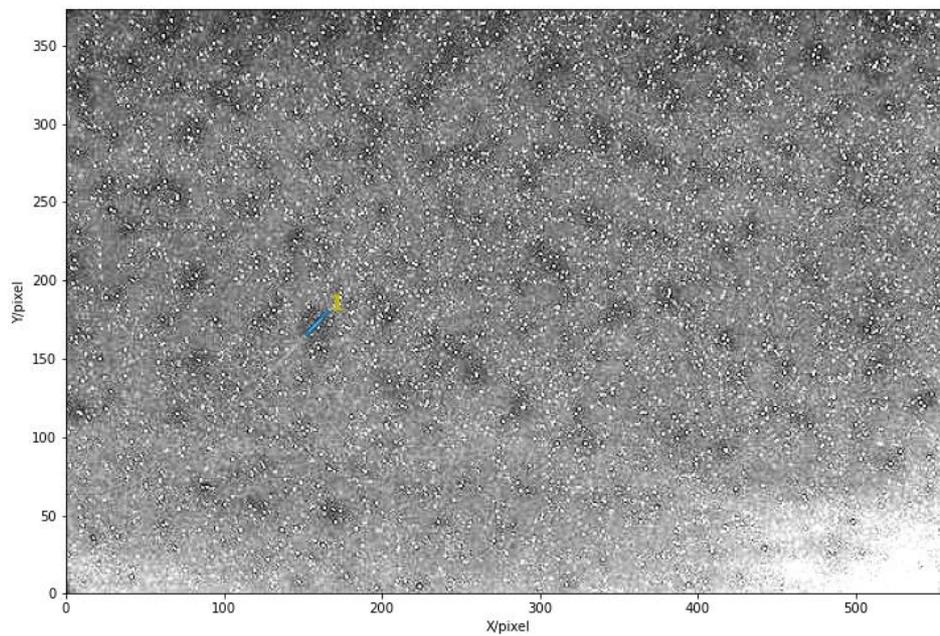
These pictures are in image format, *PNG*, or *JPG*. However, ASTRiDE only reads *FITS* format. So, they are converted using the *GIMP* software, which is a platform for image edition. After that, these files were stored in the same folder of the algorithm, and ASTRiDE read one by one.

The algorithm was able to identify all the streaks in every image. Two pictures were chosen to show the results, one with a very short streak and the other with a pretty large one. This last one is an image of the STARLINK satellites. These pictures will be denominated from now on as Short and Long, respectively.

The following Figures 24, and 25, and Table 10 show the results for the Short. Figure 24 is the original image, and Figure 20 is the output from ASTRiDE.



*Figure 24 Short: original image*



*Figure 25 Short: output from ASTRiDE. The streak was identified as 1*

Table 10 Streak parameters

<i>ID</i>	1	<i>Shape factor</i>	0.179
<i>x_center</i>	158.60	<i>Radius deviation</i>	0.55
<i>y_center</i>	172.21	<i>Slope angle</i>	45.95
<i>Area</i>	28.8	<i>Intercept</i>	8.22
<i>Perimeter</i>	45	<i>Connectivity</i>	-1

As can be seen in the figures, the streak is pretty small and, even so, the algorithm was able to detect it. Now, making the same procedure for the Long image, the results are presented below.

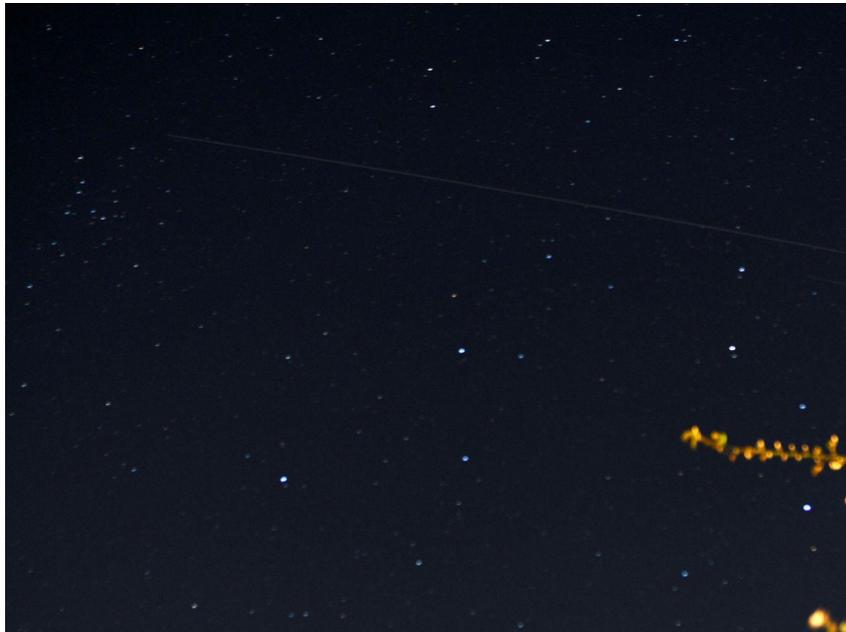


Figure 26 Long: original image

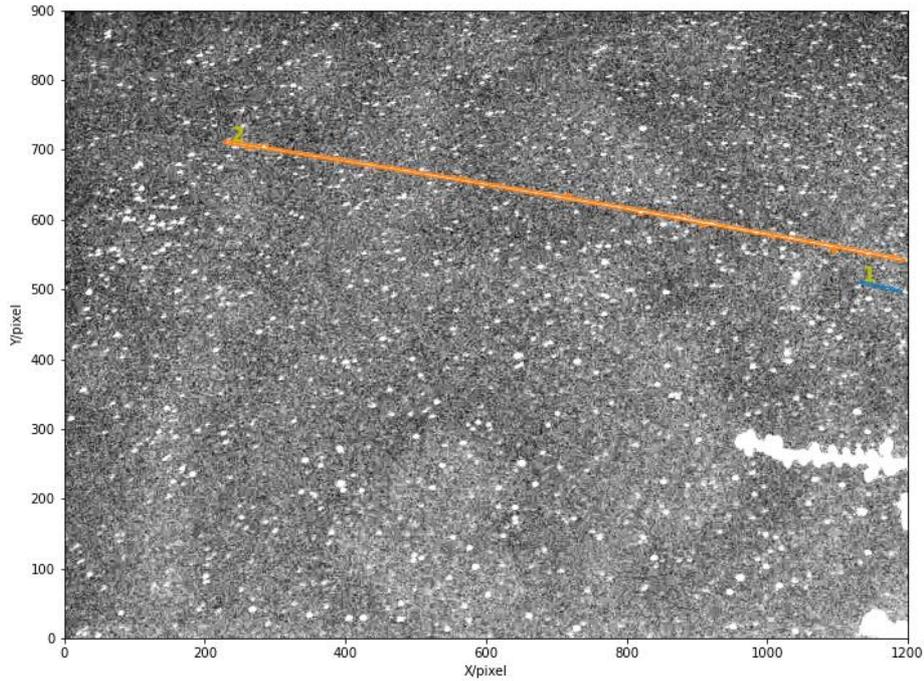


Figure 27 Long: output from ASTRiDE. Two streaks were identified and denominated as 1 and 2

The algorithm identified two streaks in the image, the blue one named 1 and the orange denominated as 2. Then, the following table is related to the parameters of these two streaks.

Table 11 Streaks 1 and 2 parameters

<i>ID</i>	1	2	<i>Shape factor</i>	0.119	0.011
<i>x_center</i>	1162.08	729.67	<i>Radius deviation</i>	0.56	0.58
<i>y_center</i>	503.93	626.76	<i>Slope angle</i>	-12.74	-9.96
<i>Area</i>	166	3955.4	<i>Intercept</i>	766.56	754.88
<i>Perimeter</i>	132.5	2132.9	<i>Connectivity</i>	-1	-1

Streak 2 is clearly larger than 1, and this is confirmed in Table 11, where the area and the perimeter of 2 are greater than 1. Now comparing the images, Long and Short, the streak of the Short is visually smaller than the two in the Long, and the results verify that.

**Outcome**

ASTRiDE was able to identify several types of streaks, short, long, faint, and bright. The algorithm is also very well explained in Github, which makes it easy to use. The only disadvantage is that it only reads *FITS* files, but this is simple to handle. Therefore, ASTRiDE can be applied in this work.

## 6 Probability of space debris detection

The detection of space debris in an optical image is related to the fact of the object to be or not in the picture. Thus, this chapter studies the probability of detection of space debris. First, it is necessary to know the size of the object and the amount in orbit. So the visual magnitude is used to calculate the minimum size that the image sensor can detect, and the quantity of debris is determined by applying a power law. Then, knowing these values, it is possible to calculate the probability of detection.

### 6.1 Using visual magnitude to determine the size of debris

The determination of the size from the visual magnitude was done using equation 2.5. In that equation, information about the solar phase angle and the mixing coefficient of the object is needed. So, a Python program was done to simulate different values of these parameters and evaluate the results as a function of these two amounts.

As said in chapter 2, the debris is considered a sphere, and the albedo is 0.175. The program starts to calculate the cross-section area from the radius defined by the user. The distance between the object and the observer is a user input as well. After that, the solar phase angle function specular and diffuse are calculated with  $\phi$  ranging from 0 to 180 deg. Then, three values of the mixing coefficient were chosen:  $\beta$  equal to 0, 0.5, and 1. The following figure shows the results for a 10 cm debris in an 800 km orbit.

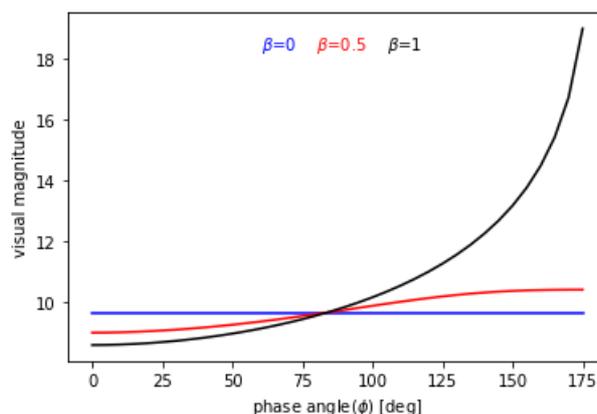


Figure 28 Visual magnitude for three different mixing coefficients for a 10 cm debris in an 800 km orbit

These results indicate the minimal visual magnitude a sensor must have to detect space debris of this size in a specific orbit. In chapter 4, it was calculated a magnitude of 7 to the image sensor. So, it is predicted to detect debris larger than 40 cm in an 800 km orbit, 10 cm at 200 km, and 1 m at 2000 km.

Besides that, satellites with know magnitudes were simulated to validate the model developed. However, it is hard to find and predict the albedo of the satellites, so the calculations were done for the ones that have this parameter available.

According to Horiuchi et al. (2020), the satellite *STARLINK-1113* has an albedo of about 0.04, and the *STARLINK-1130*, know as *Darksat*, has an approximate albedo of 0.037. These two satellites can be assumed to be a sphere of 1.5 m of radius, and they are in a 550 km orbit.

Another satellite considered in the calculations was the *NOAA-20*, which has an albedo of about 0.25. It is in an 830 km orbit and can be assumed to be a sphere of 1.5 m of radius (Hainaut & Williams, 2020). Furthermore, the observed magnitude of satellites is available online at <http://mmt9.ru/satellites/>, where it is presented the statistical data of these observations with information of the minimal, maximum, and median magnitudes. The comparison of the simulated satellites' magnitudes with the observational values is indicated in Table 12. The calculations were done considering the phase angle varying from 0 to  $\pi$ . Also, the same three values for the mixing coefficient were chosen:  $\beta$  equal to 0, 0.5, and 1.

Table 12 Comparison of the satellites' simulated and observational magnitudes

<i>Satellites</i>	$0 \leq \phi \leq \pi$	<i>Observed magnitude</i>		
	$m_{mean}$	$m_{min}$	$m_{max}$	$m_{median}$
<i>STARLINK-1113</i>	4.95	4.38	6.38	6.02
<i>Darksat</i>	5.04	2.91	7.94	6.17
<i>NOAA-20</i>	3.86	3.14	5.76	4.9

The simulated mean magnitudes in Table 12 are calculated, taking into account the three different values for the mixing coefficient.

The comparison among the values shows that the simulated magnitudes have results between the minimal and maximum observed. However, these outputs are closer to the minimal ones, revealing that the calculations tend to represent the brighter values of the objects. This was expected due to all the approximations. Nevertheless, the model proves that it is capable of providing reliable results.

## 6.2 Power law

By the catalog, the last estimation of the amount of space debris over 10 cm in *LEO* is 16464 (ESA, 2020a). The sensor of this work can detect these debris sizes since the previous model predicted that the minimal size detectable is 10 cm.

However, the debris was assumed to be spherical in the magnitude calculations. This is not the case for the catalog. So, a model was developed to provide a number of objects in orbit, supposing they are spherical. This calculation was based on a model that determines the number of objects beyond Saturn using the power law (Luu & Jewitt, 1988).

Power law distributions are used to describe the frequency of occurrences of meteoroid masses, the number of bodies somewhere in space, stellar masses, crater diameters, and the size distribution of objects. The power law is a function of the independent variable (the debris radius for this work) and two other parameters, the power law index and the power law constant (Colwell, 1993).

There are two types of power law, differential and cumulative distribution. Integrating the differential distribution provides the cumulative. Power law differential distribution can be written as:

$$n(r)dr = C r^{-\alpha} dr \quad (6.1)$$

In equation 6.1,  $r$  is the debris radius,  $C$  is the power law constant, and  $\alpha$  is the power law index. The cumulative distribution has the format of:

$$n(r) = C_c r^{-b} \quad (6.2)$$

where  $b = \alpha - 1$  and  $C_c = C/b$ .

Following the work done by Luu & Jewitt (1988), the expression for the number of debris with radius in the range  $r$  to  $r + dr$  having a distance from the Earth equal to  $R$  is

$$dN = n(r) \frac{4}{3} \pi R^3 dr = C r^{-\alpha} \frac{4}{3} \pi R^3 dr \quad (6.3)$$

Considering the *LEO* orbit as a spherical shell with  $R_{min} = 200 \text{ km}$  e  $R_{max} = 2000 \text{ km}$ , the equation 6.3 can be integrated with a given interval for the debris radius.

It was chosen an  $r$  ranging from 10 cm to 1 m.

$$N = \frac{4}{3} \pi C (R_{max}^3 - R_{min}^3) \int_{r_{min}}^{r_{max}} r^{-\alpha} dr \quad (6.4)$$

To proceed, it is necessary the values of  $\alpha$  and  $C$ . However, this information was not found in the literature. So, these parameters were calculated using data of the number of debris for a given size in the *LEO* orbit, Table 13.

Table 13 Number of debris for different sizes at the reference epoch of May 2009.  
Adapted from (Bonnal & McKnight, 2017)

<i>Diameter</i>	<i>&gt; 1 cm</i>	<i>&gt; 10 cm</i>	<i>&gt; 1 m</i>
<i>Number of debris</i>	744084	29210	5442

More recent data was not found. This information was plotted in a graphic following the power law cumulative distribution equation to extract the values of the parameters. Thus, the results obtained were:  $b = 1.4$  and  $C_c = 3.5 \times 10^{-17}$ . Now, applying the relation between the cumulative and differential distribution,  $\alpha = 2.4$  and  $C = 4.9 \times 10^{-17}$ . Finally, substituting the values of  $\alpha$  and  $C$  into equation 6.4 results in 9032 debris inside the *LEO* orbit. Considering that the number of catalog debris of the same reference epoch, 2009, was 9134 (Yurasov & Shargorodskiy, 2009), the power law's result is satisfactory.

### 6.3 Probability of space debris detection

Some assumption was made to calculate the probability of detection. First of all, the sky was considered to be a sphere known as the celestial sphere. Astronomers do this technique to position astronomical objects.

The celestial sphere is assumed to have a unitary radius and a complete area of  $4\pi$  steradians. Steradian is the Standard International (SI) unit of solid angular measure. In other words, steradians are related to a sphere's surface area in the same way a radian is associated with the circumference of a circle. It is also possible to convert from steradian to square degrees as follows:

$$1 \text{ steradian} = \left(\frac{180}{\pi}\right)^2 \quad (6.5)$$

Then, the celestial sphere has  $41252.96 \text{ deg}^2$ . Considering that the 9032 debris found previously are uniformly distributed in the celestial sphere, and an image sensor capable of observing  $1 \text{ deg}^2$  of the sky, the probability of an object be inside the observational area is:

$$P = \frac{9032}{41252.96} \cong 0.22 \quad (6.6)$$

However, the debris is moving in orbit, so this probability is not fixed, and the chance to detect an object increase with higher exposure times. Then, the probability of detection related to the time can be calculated by the binomial distribution, which is associated with only two results, success or failure (Correa, 2003). In this work, a successful event means at least one debris detection.

The binomial probability has the following properties:

- The experiment consists of  $n$  repeated trials;
- Each trial can result in the two possible outcomes, success or failure;
- The probability of success is the same in every trial;
- The trials are independent.

The binomial distribution is calculated by

$$B(X = x) = \binom{n}{x} P^x Q^{n-x} \quad (6.7)$$

Where  $x$  is the number of success,  $n$  is the number of trials,  $P$  is the probability of success on an individual trial, and  $Q$  is the probability of failure in an individual trial ( $Q = 1 - P$ ).

The number of successes is equal to 1, that is, at least one debris detected. The probability of success is the value found in equation 6.6,  $P = 0.22$ . The number of trials is related to the exposure time and the angular velocity ( $\omega$ ) of the debris,  $n = \text{exposure time} \times \omega$ .

The angular velocity of the debris can be calculated by knowing the velocity in a specific orbit. According to Culp (1991) and Schildknecht (2007), the velocity ( $v$ ) of space debris in the *LEO* orbit is between 7 and 8 km/s. So, consider objects with these two velocities in an 800 km orbit and three different exposure times, the probabilities of detecting at least one debris in each event using equation 6.7 are summarized in Table 14.

Table 14 Detection probabilities of space debris in 800 km orbit for different exposure times and velocities

Exposure time [s]	$v = 7 \text{ km/s} \rightarrow \omega = 0.50 \text{ deg/s}$		$v = 8 \text{ km/s} \rightarrow \omega = 0.57 \text{ deg/s}$	
	n	B(X = 1)	n	B(X = 1)
2	1	0.22	1	0.22
10	5	0.71	6	0.77
100	50	1	57	1

In table 14, it can be noted that the probability increases with the exposure time. With 100 seconds, the chance to have at least one debris inside the  $1 \text{ deg}^2$  is 100%.

The same calculation can be done considering the field of view obtained for the image sensor in the telescope in chapter 4. It was chosen the FOV of the optical system sensor and telescope allows pointing to a specific region of the sky.

The FOV can be approximated to be equal to  $20 \text{ arcminutes}^2$  ( $FOV \cong 5 \times 4 \text{ arcminutes}$ ). Therefore, the probability to find an object inside this observational area is:

$$P \cong 0.00122 \quad (6.8)$$

Using the previous consideration (objects with velocities between 7 and 8 km/s in an 800 km orbit), the probabilities of detecting at least one debris for different exposure times are shown in the following table.

Table 15 Detection probability using the FOV of the image sensor in the telescope

<i>Exposure time [s]</i>	<i>v = 7 km/s → ω = 0.50 deg/s</i>		<i>v = 8 km/s → ω = 0.57 deg/s</i>	
	n	B(X = 1)	n	B(X = 1)
<i>2</i>	1	0.00122	1	0.00122
<i>10</i>	5	0.0061	6	0.0073
<i>100</i>	50	0.059	57	0.067
<i>1000</i>	501	0.45	573	0.5
<i>5000</i>	2507	0.95	2865	0.97

These results demonstrate that the probability of detection decrease with the reduction of the observational area. However, it is still achievable to reach a 100% probability of detecting space debris, but it needs more observation time.

Even so, it is possible a non-detection of the space debris by the image sensor because it also depends on the visual magnitude. If the object's brightness is less than the limit magnitude of the sensor, it will not be observed.

### 6.3.1 Probability of detection in a specific orbit

The previous calculations were done assuming uniformly distributed debris in the whole celestial sphere. However, it has already been seen in chapter 2 that this is not true. So, for more accurate results, it was considered the most crowded region in the *LEO* orbit. According to Bonnal & McKnight (2017), the greater debris concentration is at an altitude between 775 and 825 km and an orbit inclination (i) among 100 and 102 degrees. The debris in this region represents 9% of the total. These orbits with higher inclination are known as polar orbit, and it is commonly used to investigate weather patterns (ESA, 2020b).

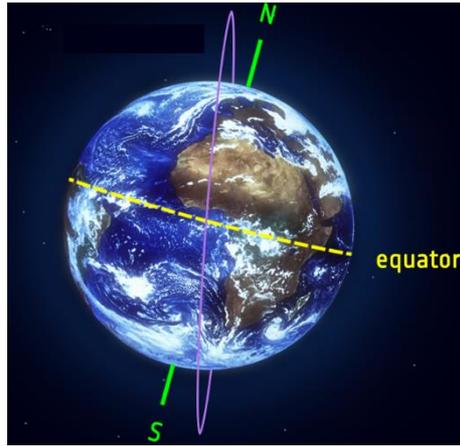


Figure 29 Polar orbit (ESA, 2020b)

The same approach was considered to determine the probability of detection. Assuming the sky as a celestial sphere, now it is necessary to calculate the area between 100 and 102 degrees. Considering the spherical coordinates:

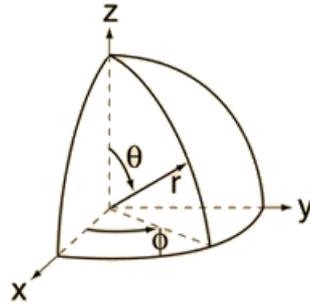


Figure 30 Spherical coordinates

The problem is reduced to calculate the area of a surface of 2 deg. Recalling that the celestial sphere has a unitary radius, then the surface area can be determined as

$$S = \iint \sin \theta \, d\theta \, d\phi \quad (6.9)$$

Since the area of interest is a region of 2 deg, the zone between 89 and 91 deg was chosen for  $\theta$  to facilitate the calculation. These angles were transformed into radians, and as  $0 \leq \phi \leq 2\pi$ , the surface area is

$$S \cong 0.22 \text{ steradians} \cong 719.96 \text{ deg}^2 \quad (6.10)$$

The altitude of this area is between 775 and 825 km. Then considering the mean, 800 km, it is possible to repeat the calculation for the optical system FOV to the new area. Since there is 813 debris in this region, 9% of the total, and assuming they are uniformly distributed, the probability of detecting an object inside the FOV in this zone is  $P \cong 0.0063$ .

Considering debris at the same velocity as before,  $7 \text{ km/s} \leq v \leq 8 \text{ km/s}$ , then the probabilities of detecting at least one object are summarized in Table 16.

Table 16 Detection probability in the most crowded area using the FOV

Exposure time [s]	$v = 7 \text{ km/s} \rightarrow \omega = 0.50 \text{ deg/s}$		$v = 8 \text{ km/s} \rightarrow \omega = 0.57 \text{ deg/s}$	
	n	B(X = 1)	n	B(X = 1)
2	1	0.0063	1	0.0063
10	5	0.031	6	0.037
100	50	0.27	57	0.3
500	251	0.79	286	0.83
1000	501	0.96	573	0.97

According to Table 16, it is evident that orbit selection is another vital parameter to detect debris. Therefore, the probability of observing at least one object in the most crowded area with an exposure time of 1000 s is almost 100%.

### 6.3.2 Probability of detection between latitudes

Another approach can be considered using latitude. While the inclination is a parameter related to the angle between the equator and the orbit plane, being fixed to a specific orbit, the latitude (L) is a varying parameter because it measures the angle between the parallel plane of the moving object in orbit and the equator. In one latitude is presented a wide range of orbits with different inclinations. The maximum value for the latitude is 90 deg and the minimum,  $-90 \text{ deg}$ .

It is expected to be more likely to find objects in higher latitudes. So, the following calculations are done to prove this concept. Considering orbits of 100 deg inclination, values larger than 90 deg indicate that the object's movement is retrograde. Then, the only difference between the orbits of 100 deg and 80 deg of inclinations is the sense of direction. Thus, the last one can be chosen to facilitate the calculations.

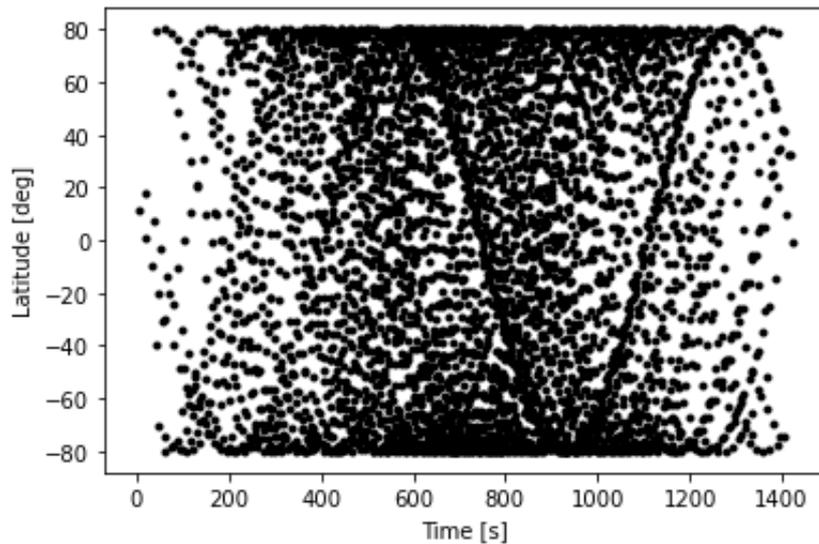
In an orbit of  $i = 80 \text{ deg}$ , the debris latitude ranges from  $-80$  to  $80 \text{ deg}$ . Still, the orbit of the objects is a periodic function. In a 2D graph, the trajectory has a sinusoid behavior. So, the orbit can be described as:

$$x = 80 \cos\left(\frac{2\pi}{T}t - \varphi\right) \quad (6.11)$$

Where  $\varphi$  is the phase,  $T$  is the period of the orbit,  $t$  is the time, and 80 is the amplitude, the maximum value for the latitude. This orbit is an 800 km altitude, and, considering objects with a velocity of 7 km/s, the period can be calculated.  $T$  has a value equal to 718.10 s.

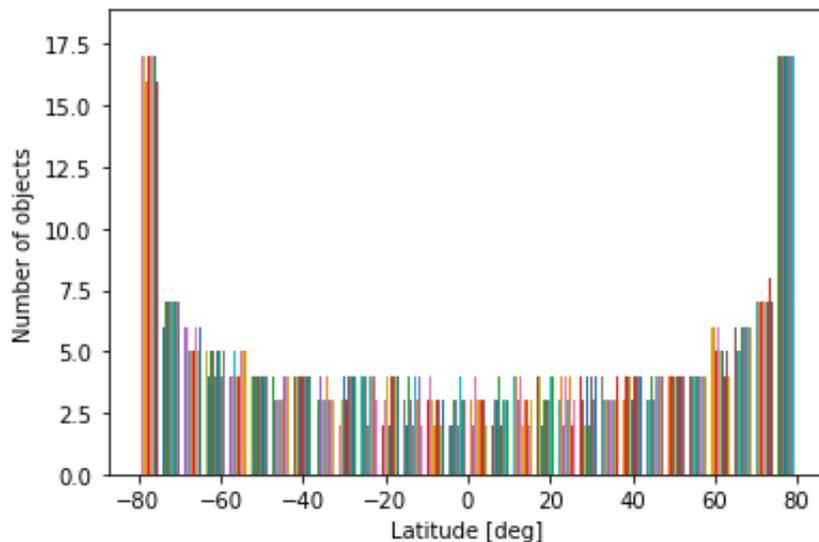
To simulate several objects with different trajectories, it was done a Python code that generates random values for the initial time, ranging from 0 to T, and for the phase, varying between 0 and  $2\pi$ . With these values, the debris orbit will be represented with a time interval ranging from the initial time until it finalizes a complete revolution.

A hundred random values were generated to the initial time and phase, symbolizing 100 different debris trajectories. This result is shown in Figure 31. It can be noted a larger concentration of objects in higher latitudes.



*Figure 31 A hundred different debris trajectories in an orbit of latitude between  $-80$  and  $80$  deg*

These results are more evident in a histogram, Figure 32. So, simulating a thousand random values, it is possible to see that the objects are more likely to be in higher latitudes as expected.



*Figure 32 Histogram for a thousand different debris trajectories in an orbit of latitudes between  $-80$  and  $80$  deg*

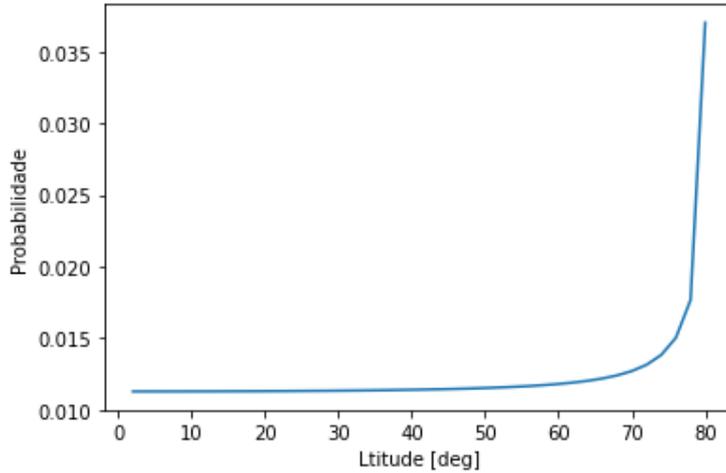
According to Chobotov (2007), the probability of finding an object between latitude  $-L$  and  $L$  is

$$P(-L, L) = \frac{2}{\pi} \sin^{-1} \left( \frac{\sin L}{\sin i} \right) \quad (6.12)$$

The orbits with inclinations equal to  $80$  deg are kept to the calculations. In equation 6.12, the probability is for a large range of latitudes. For results inside a smaller band of latitudes, for example, objects in latitudes from  $78$  deg to  $80$  deg, the probability has to be calculated as

$$P(78, 80) = \frac{1}{2} (P(-80, 80) - P(-78, 78)) \quad (6.13)$$

Equation 6.13 is divided by two to take only a hemisphere of the celestial sphere. Then, the probabilities of finding objects inside a band of  $2$  deg for latitudes ranging from  $0$  to  $80$  deg are shown in Figure 33.



*Figure 33 Probabilities of finding objects inside a band of  $2$  deg for different latitudes*

Figure 33 also confirms that it is more expected to find objects in higher latitudes. The probability of an object be inside a band between  $78$  and  $80$  deg is  $0.037$ . This result indicates that  $334$  debris from the total,  $9032$ , are in this area.

The probability of detection is done using the same approach as the previous calculation. First, the surface area of the region between these two latitudes is determined. The area can be calculated by equation 6.9 considering  $10 \text{ deg} \leq \theta \leq 12 \text{ deg}$  and  $0 \leq \phi \leq 2\pi$ .

$$S \cong 0.042 \text{ steradians} \cong 137.37 \text{ deg}^2 \quad (6.14)$$

The 334 objects are inside this area. The probability of detecting an object in the FOV in this zone is  $P \cong 0.0135$ . Considering debris at the same velocity as before,  $7 \text{ km/s} \leq v \leq 8 \text{ km/s}$ , then the probabilities of detecting at least one object are summarized in Table 17.

*Table 17 Detecting probability for at least one object in latitudes between 78 and 80 deg*

<i>Exposure time [s]</i>	<i>v = 7 km/s → ω = 0.50 deg/s</i>		<i>v = 8 km/s → ω = 0.57 deg/s</i>	
	n	B(X = 1)	n	B(X = 1)
<i>2</i>	1	0.0135	1	0.0135
<i>10</i>	5	0.0657	6	0.0783
<i>100</i>	50	0.493	57	0.539
<i>500</i>	251	0.967	286	0.979

Comparing Tables 16 and 17, it is possible to see that it is more likely to detect at least one object between latitudes than considering an inclination.

### **Outcome**

This chapter proved that there is a big chance to detect space debris by the image sensor. It is only necessary to choose a reasonable exposure time and, if the object has enough brightness and size, it will be observed.

## **7 Tests and results**

## **8 Conclusions**

## 9 References

- Andrenucci, Pergola, & Ruggiero. (2011). Active Removal of Space Debris Expanding foam application for active debris removal. *ESA Final Report*, 1–132.
- Bonnal, C., & McKnight, D. (2017). *IAA Situation Report on Space Debris - 2016*.
- Brown, J., Stubis, K., & Cahoy, K. (2017). TETRA: Star Identification with Hash Tables. *31st Annual AIAA/USU Conference on Small Satellites*, 831, 1–12. <https://github.com/brownj4/Tetra>
- Chobotov, V. A. (2007). Orbital mechanics. In *Modeling and Simulation in Science, Engineering and Technology* (Vol. 27). [https://doi.org/10.1007/978-0-8176-4438-3\\_5](https://doi.org/10.1007/978-0-8176-4438-3_5)
- Clemens, S. (2019). *On-Orbit Resident Space Object (RSO) Detection Using Commercial Grade Star Trackers*. York University.
- Colwell, J. E. (1993). *Power-law confusion: you say incremental, I say differential*.
- Correa, S. M. B. B. (2003). Distribuições Discretas de Probabilidade. In *Probabilidade e Estatística* (2<sup>a</sup> ed, pp. 82–83). PUC Minas Virtual.
- Culp, R. D. (1991). Orbital debris. In *Advances in the Astronautical Sciences* (Vol. 74). <https://doi.org/10.17226/4765>
- ESA. (1997). *Hipparcos*. <https://www.cosmos.esa.int/web/hipparcos>
- ESA. (2009). *MASTER - Space Debris User Portal*. <https://sdup.esoc.esa.int/master/>
- ESA. (2020a). *About space debris*. [https://www.esa.int/Safety\\_Security/Space\\_Debris/About\\_space\\_debris](https://www.esa.int/Safety_Security/Space_Debris/About_space_debris)
- ESA. (2020b). *Type of orbits*. [https://www.esa.int/Enabling\\_Support/Space\\_Transportation/Types\\_of\\_orbits#LEO](https://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits#LEO)
- Ettouati, I., Mortari, D., & Pollock, T. (2006). Space surveillance using star trackers. Part I: Simulations. *Advances in the Astronautical Sciences*, 124 II(January 2006), 2073–2087.
- Groves, P. D. (2013). Chapter 2: Coordinate Frames, Kinematics, and the Earth. In *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems* (Second edi, pp. 23–78). Antech House.
- Gutiérrez, S. T., Fuentes, C. I., & Díaz, M. A. (2020). Introducing SOST: An ultra-low-cost star tracker concept based on a raspberry pi and open-source astronomy software. *IEEE Access*, 8, 166320–166334.

<https://doi.org/10.1109/ACCESS.2020.3020048>

- Hainaut, O. R., & Williams, A. P. (2020). Impact of satellite constellations on astronomical observations with ESO telescopes in the visible and infrared domains. *Astronomy and Astrophysics*, 636. <https://doi.org/10.1051/0004-6361/202037501>
- Hickson, P. (2018). A fast algorithm for the detection of faint orbital debris tracks in optical images. *Advances in Space Research*, 62(11), 3078–3085. <https://doi.org/10.1016/j.asr.2018.08.039>
- Horiuchi, T., Hanayama, H., & Ohishi, M. (2020). Simultaneous Multicolor Observations of Starlink’s Darksat by the Murikabushi Telescope with MITSuME. *The Astrophysical Journal*, 905(1), 3. <https://doi.org/10.3847/1538-4357/abc695>
- Hostetler, J., & Cowardin, H. (2019). *Experimentally-Derived Phase Function Approximations in Support of the Orbital Debris Program Office. 1.*
- Howell, S. B. (2006). *Handbook of CCD Astronomy* (Second edi). Cambridge University Press.
- Jain, U. (2016). *Characterization of CMOS Image Sensor*. Delft University of Technology.
- Kervin, P. W., Bolden, M., & Toth, J. (2010). Phase angle: What is it good for? *Proceedings from Advanced Maui Optical and Space Surveillance Technologies Conference*, 11. <http://www.amostech.com/TechnicalPapers/2010/NROC/Kervin.pdf>
- Kim, D.-W. (2016). ASTRiDE: Automated Streak Detection for Astronomical Images. *Astrophysics Source Code Library, ascl: 1605.009*.
- Lang, D., Hogg, D. W., Mierle, K., Blanton, M., & Roweis, S. (2010). Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *Astronomical Journal*, 139(5), 1782–1800. <https://doi.org/10.1088/0004-6256/139/5/1782>
- Lopes, S. F. (2018). *Development of a low-cost Star Tracker for Cubesats*. Universidade de Lisboa.
- Luu, J. X., & Jewitt, D. (1988). A Two-Part Search for Slow-Moving Objects. *The Astronomical Journal*, 95, 8.
- Miguel, N. (2005). *Physical and Chemical Characterization of the Trans-Neptunian Objects Population*. Universidade de Lisboa.
- Pedrotty, S., Lovelace, R., Christian, J., Renshaw, D., & Quintero, G. (2020). Design and Performance of an Open-Source Star Tracker Algorithm on Commercial Off-The-Shelf Cameras and Computers. *43rd Annual AAS Guidance, Navigation, and*

- Control Conference, February, 1–12.*  
<https://www.researchgate.net/publication/339076698>
- Romanishin, W. (2002). An Introduction to Astronomical Photometry Using CCDs. *Image (Rochester, N.Y.)*, 167. <http://observatory.ou.edu>
- Rossi, A. (2005). Population models of space debris. *International Astronomical Union*. <https://doi.org/10.1017/S1743921304008956>
- Sarvi, M. N., Abbasi-Moghadam, D., Abolghasemi, M., & Hoseini, H. (2020). Design and implementation of a star-tracker for LEO satellite. *Optik*, 208(December 2019), 164343. <https://doi.org/10.1016/j.ijleo.2020.164343>
- Schildknecht, T. (2007). Optical surveys for space debris. *Astronomy and Astrophysics Review*, 14(1), 41–111. <https://doi.org/10.1007/s00159-006-0003-9>
- Spratling, B. B., & Mortari, D. (2009). A survey on star identification algorithms. *Algorithms*, 2(1), 93–107. <https://doi.org/10.3390/a2010093>
- Tennenbaum, A. (2017). Automatic Star-tracker Optimization Framework. *31st Annual AIAA/USU Conference on Small Satellites*, 1–5. <http://openstartracker.org/>
- US Space Surveillance Network*. (n.d.). Retrieved April 2, 2021, from <https://www.stratcom.mil/Media/Factsheets/Factsheet-View/Article/976414/usstratcom-space-control-and-space-surveillance/>
- Vananti, A., Schild, K., & Schildknecht, T. (2015). Streak detection algorithm for space debris detection on optical images. *AMOS Technical Conference*, 15–18.
- Virtanen, J., Poikonen, J., Säntti, T., Komulainen, T., Torppa, J., Granvik, M., Muinonen, K., Pentikäinen, H., Martikainen, J., Näränen, J., Lehti, J., & Flohrer, T. (2016). Streak detection and analysis pipeline for space-debris optical images. *Advances in Space Research*, 57(8), 1607–1623. <https://doi.org/10.1016/j.asr.2015.09.024>
- Vitiello, F. (2020). Performance Analysis of Space Surveillance Using Space Based Optical Sensors. *Aerotecnica Missili & Spazio*, 99(4), 263–273. <https://doi.org/10.1007/s42496-020-00063-1>
- Waltham, N. (2013). CCD and CMOS sensors. *Observing Photons in Space, Ccd*, 423–442. [https://doi.org/10.1007/978-1-4614-7804-1\\_23](https://doi.org/10.1007/978-1-4614-7804-1_23)
- Xi, J., Wen, D., Ersoy, O. K., Yi, H., Yao, D., Song, Z., & Xi, S. (2016). Space debris detection in optical image sequences. *Applied Optics*, 55(28), 7929. <https://doi.org/10.1364/ao.55.007929>
- Yurasov, V. S., & Shargorodskiy, V. D. (2009). Features of space debris survey in LEO utilizing optical sensors. *European Space Agency, (Special Publication) ESA SP*,

672 SP(July).